Cooperative multi-agent system for coordinated traffic signal control

D. Srinivasan and M.C. Choy

Abstract: A novel, distributed, cooperative multi-agent approach employing multiple, interacting, autonomous hybrid agents to provide effective signal control for real-time traffic management is presented here. This multi-agent system uses advanced cooperative behaviours to improve individual agents' learning process and adaptability. Each agent cooperates with other agents within its cooperative zone, the size of which changes dynamically according to the changing needs of the agent. The performance of the proposed cooperative ensemble multi-agent system is tested on a large, complex traffic network and compared against two other approaches. For the 6 h extreme scenario with two peaks, the proposed approach reduces the total mean delay by 35.6% when compared to the GLIDE benchmark, while for the 24 h extreme scenario with multiple peaks, the reduction is 75%. The results demonstrate the efficacy of the cooperative multi-agent approach in dealing with the approximated version of an infinite horizon dynamic problem.

1 Introduction

The increase in urbanisation and traffic congestion create an urgent need to operate our transportation systems with maximum efficiency. Real-time traffic signal control is an integral part of modern Urban Traffic Control Systems aimed at achieving optimal utilisation of the road network. Providing effective real-time traffic signal control for a large complex traffic network is an extremely challenging distributed control problem. Signal system operation is further complicated by the recent trend that views traffic signal system as a small component of an integrated multimodal transportation system. Optimisation of traffic signals and other control devices for the efficient movement of traffic on streets and highways constitutes a challenging part of the advanced traffic management system of intelligent transportation system. For a large-scale traffic management system, it may be difficult or impossible to tell whether the traffic network is flowing smoothly and assess its current state. In addition, due to some of the nonlinear and stochastic traffic processes in a traffic network, predicting the effects of modifying any of the traffic control parameters is not easy.

Over the years, researchers have applied various artificial intelligence techniques such as neural network [1], fuzzy logic [2–5], genetic algorithms as well as multi-agent systems [6–10] to develop better traffic signal control systems. Based on these research works, various advantages of using the different computational intelligence techniques can be discerned and employed to improve some of the existing architectures. However, it is evident that more needs be done to implement a systematic, unsupervised, distributed control scheme for testing in a complex traffic network simulating a real world scenario.

Over the past few years, multi-agent systems have become a crucial technology for effectively exploiting the increasing availability of diverse, heterogeneous and distributed information sources. Researchers over the years have adopted numerous techniques and used various tools to implement multi-agent systems for their problem domains. As researchers gain a better understanding of these autonomous multi-agent systems, more features are incorporated into them to enhance their performance and the enhanced systems can then be used for more complex application domains.

The research work in [11] uses a multi-agent system that integrates the Simultaneous Perturbation Stochastic Approximation (SPSA) technique and fuzzy neural networks (NNs). This system [11] has been successfully tested in a simulated traffic network and is shown to perform better than a simulated GLIDE benchmark while overcoming two limitations of the research work reported in [12]. Nevertheless, it has been observed that the multi-agent system in [11] lacks advanced cooperative behaviours, as each agent may not be able to gain a higher perspective of the overall problem given that each of them is limited to cooperate within a pre-specified, fixed cooperative zone. In this paper, we present a new cooperative ensemble (CE) of intelligent agents which are equipped with advanced cooperative behaviours. Each agent in the CE is able to dynamically determine the size of its cooperative zone. Cooperation within the cooperative zone takes the form of lateral information exchange and negotiation. The proposed CE-based multi-agent system was evaluated against the SPSA–NN based multi-agent system [11] and the simulated GLIDE benchmark using the simulation scenarios presented in [11].

2 Distributed control problem for the CE

A large scale control problem such as controlling the traffic signals in a traffic network can be divided up into

IEE Proceedings online no. 20055011

doi:10.1049/ip-its: 20055011

Paper received 9 December 2005

The authors are with the Department of Electrical & Computer Engineering, National University of Singapore, 10 Kent Ridge Crescent, Singapore 117576

E-mail: dipti@nus.edu.sg

sub-problems where each sub-problem is being handled by a local controller or an agent. For such a distributed approach, each agent will generate its own control variables based on the local information it receives. Also, exchange of information can be present among the different agents either laterally (for agents in the same hierarchical level) or vertically (between lower level agents and higher level agents). In this system, the various parameters can be defined as follows:

$G = (N, L)$	Directed graph with a set N of n nodes and a set L of l links describing the traffic network
A_i	The CE agent acting at node $i \in N$
T	Total number of temporal stages at time t
$I_i(T)$	Local input information vector of A_i at stage T
$c_i(T)$	Vector of control parameters of A_i at stage T
$c_i(T)^o$	Optimal vector of control parameters of A_i at stage T
$l_{ij}(T)$	The link relationship between node i and node j as seen from node i at stage T (note that $l_{ij} \neq l_{ji}$)
$l_{ij}(T)^o$	The optimal link relationship between node i and node j as seen from node i at stage T
$\{L_i\}$	The set $\{L\}_i$ contains all l_{ij}, for $j = 1, 2, \ldots, n$, and $j \neq i$, where n is the total number of nodes in G
$u_i(T) = F_{iT}[I_i(T), c_i(T)]$	Control function of A_i at stage T
$u_i(T)^o = F_{iT}^o[I_i(T), c_i(T)^o]$	Optimal control function of A_i at stage T
$s(T)$	State vector of the traffic network at stage T
C	Cost function for the entire traffic network

The distributed control problem of the traffic network can thus be rephrased as:

Find the set of control function $u_i(T)$ and $\{L_i\}$ for each A_i, $i \in N$ that minimizes the cost function C, where C is a function of the states of the traffic network at different temporal stages.

In the case where all the A_i of the traffic network are designed for continuous traffic signal control, the optimization problem becomes that of an infinite-horizon one. As can be inferred from the above, the link relationship and the control function are the two main factors which can affect the agent's choice of traffic signal control policy.

The proposed multi-agent system uses CEs of distributed agents to address this optimization problem. The following section describes the architecture of the CE as well as the structure of each agent.

3 The architecture of the CE

The CE is a single layer multi-agent system where each agent is assigned to control the traffic signal of a signalised intersection or node in the traffic network. The main features of this multi-agent system using CE of agents are as follows:

(1) Each agent in the CE has its own cooperative zone. A cooperative zone of an agent A_i will contain a set of other agents A_j, where $j \neq i$, whose links L are deemed to be 'closed' enough to the node of A_i. As such, A_i will approximate its optimal control directives based on its own perspective as well as the perspectives of the set of agents within its cooperative zone.

(2) Cooperative zones of individual agents are updated and allocated dynamically. The size of various cooperative zones, and the number of agents within each zone are allowed to vary.

(3) Although there is no physical hierarchy in the CE of agents, each agent receives different levels of perspectives based on the size of its cooperative zone.

(4) The perspectives of other agents affect the decisions of A_i in a stochastic manner.

(5) This online updating process is done continuously and each agent is expected to be able to run indefinitely in an unsupervised manner, providing effective control for the dynamically changing problem.

The CE has a single-layer, non-hierarchical multi-agent system architecture as shown in Fig. 1.

Each agent in the CE obtains input traffic parameters from the simulated traffic network at the start of every stage. Once these input traffic parameters are processed, the agents seek to control their respective nodes in a collaborative manner whenever necessary (details are provided in the next section). The various traffic signal control policies for that stage are generated next and are implemented into the simulated traffic network in real-time. This control process repeats itself again for the next stage and it essentially is an ongoing traffic control loop.

3.1 The structure of individual agent

The general schematic of agent A_i is shown in Fig. 2. Each A_i essentially generates three outputs based on the inputs it receives. The first output is the problem-specific control directive(s), which in this case is the traffic signal control directive. The second output is that of the need for cooperation and the third output is the level of cooperation.

Within each A_i, NNs are used as the framework for implementing the control directive inference engine that acts as an approximator for the traffic signal control function. NN is also used to implement the cooperation factor inference engine which generates the need as well as the level of cooperation. Fig. 3 shows the structure of a five-layered NN that forms the primary structure of each A_i.

The choice of the operators between the fuzzification layer (second layer) and the implication layer (third layer) is taken to be T-norm. The choice of the operator between the implication layer (third layer) and the consequent layer (fourth layer) is taken to be S-norm. Membership functions of the terms of the fuzzy output are singletons.

Each agent A_i generates two outputs using its neural control function $u_i(T)$ (in the form of the NN) based on the inputs $I_i(T)$ it receives at stage T. These two outputs are namely $CD_i(T)$ and $CF_i(T)$. As shown in the figure, $CD_i(T)$ is derived from the control directive inference engine while $CF_i(T)$ is derived from the cooperation factor inference engine. $CD_i(T)$ represents the control directives that are specific to different control problems.

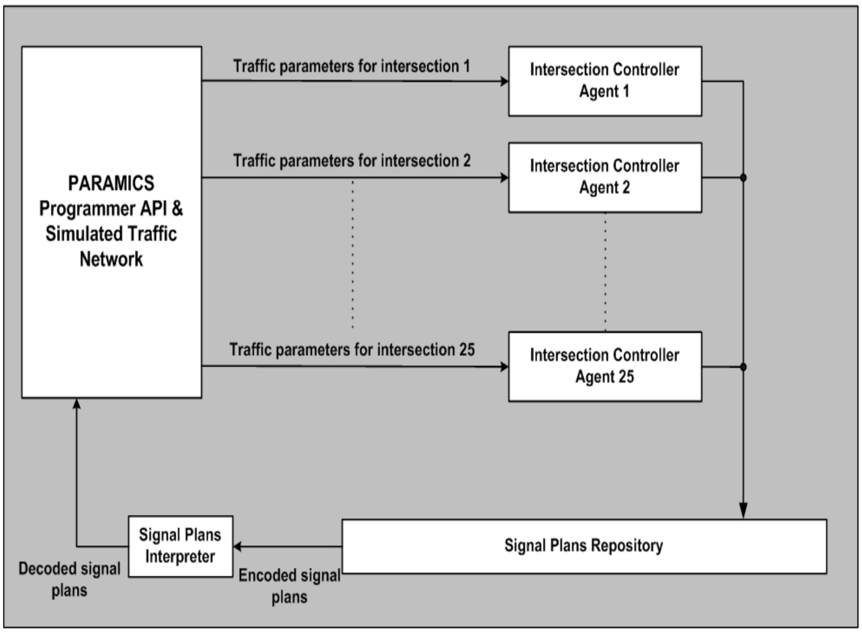

Fig. 1 *Overall structure of the multi-agent system*

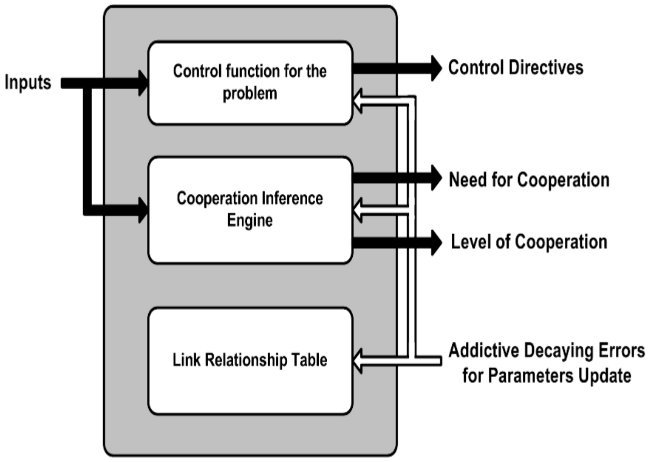

Fig. 2 *General schematic of A_i*

$CF_i(T)$ is the cooperation factor which the agent A_i needs at stage T. This is the abstraction that resembles the level of cooperation that is needed as perceived by A_i at stage T. For this multi-agent system, the level of cooperation will determine the size of the cooperative zone of A_i, which will in turn determine how the control parameters of A_i are being affected by the situations in the other nodes within A_i's cooperative zone.

3.2 Actions of each agent

The flowchart in Fig. 4 describes the sequence of actions of each agent, A_i. The various control parameters for each A_i are first initialised. Each A_i then reads in its input I_i (T) (where $T = 0$, representing the first time stage) and generates its two outputs CD_i (T) as well as CF_i (T) using its NN module described above. After all the agents have generated their outputs, an agent is randomly selected (with an equal probability of being selected) from the existing pool of agents in the multi-agent system for dynamic cooperative zone allocation. The random selection process will repeat itself until all the agents have been considered. This means that one of the following has occurred with respect to the agents:

(1) they have been selected to be part of others' cooperative zones;
(2) they have formed their own cooperative zones;
(3) they have chosen not to form any cooperative zone.

The agents within a single cooperative zone will make a group decision pertaining to their respective outputs. This means that their respective $CD_i(T)$ and $CF_i(T)$ for all A_i within that zone will be considered as a whole (either by averaging or by some other methods) and each A_i within that zone will then use the group outputs as their individual outputs. The agents who have chosen not to form any cooperative zone will continue to stick to their initial set of outputs.

Next, the set of $CD_i(T)$ for all A_i is implemented into the problem. The respective local errors are then calculated and the control parameters of each A_i updated using the cooperative parameters update algorithm (described in Section 3.5). Finally, the next set of inputs I_i $(T+1)$ will be read in for each A_i and the process will repeat itself again. There will be no stopping criterion as the problem deals with an approximated version of an infinite-horizon problem.

3.3 Dynamic cooperative zone allocation

Each agent A_i maintains a set of links $\{L_i\}$. The set $\{L_i\}$ contains l_{ij}, for $j = 1, 2, \ldots, k$, and $j \neq i$, where k is the total number of agents. The set $\{L_i\}$ for each A_i is treated as a set of control parameters, similar to the set of weights for the NN of the A_i. Hence, just like the weights, the values of every member in the set $\{L_i\}$ will be updated continuously based on the error that A_i has incurred in the previous time stage.

Fig. 5 shows the dynamic cooperative zone allocation process. When the i-th agent A_i is randomly selected for dynamic cooperative zone allocation at time stage T, it will process the various values of l_{ij} in its set $\{L_i\}$. If l_{ij} is greater than a particular threshold value T_c (to be determined empirically), A_i will then compare its $CF_i(T)$ value with the $CF_j(T)$ value of A_j. If $CF_j(T) \geq CF_i(T)$, A_j (and node n_j) will be added as part of the cooperative

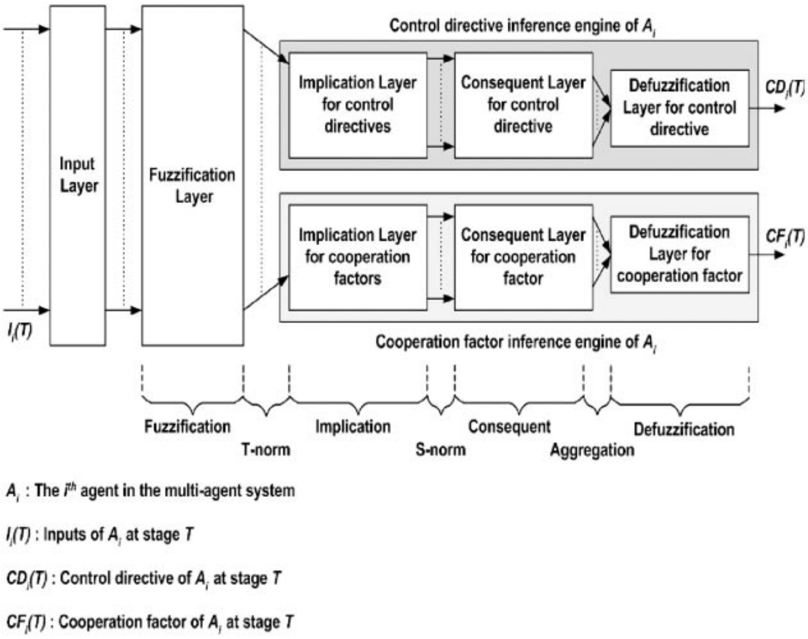

Fig. 3 *The five-layered fuzzy neural network of the CE*

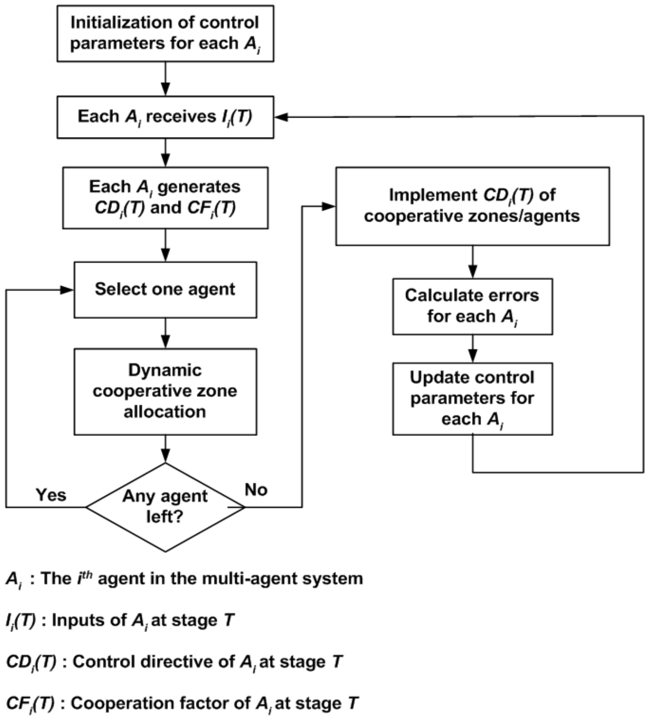

Fig. 4 *Sequence of actions taken by A_i*

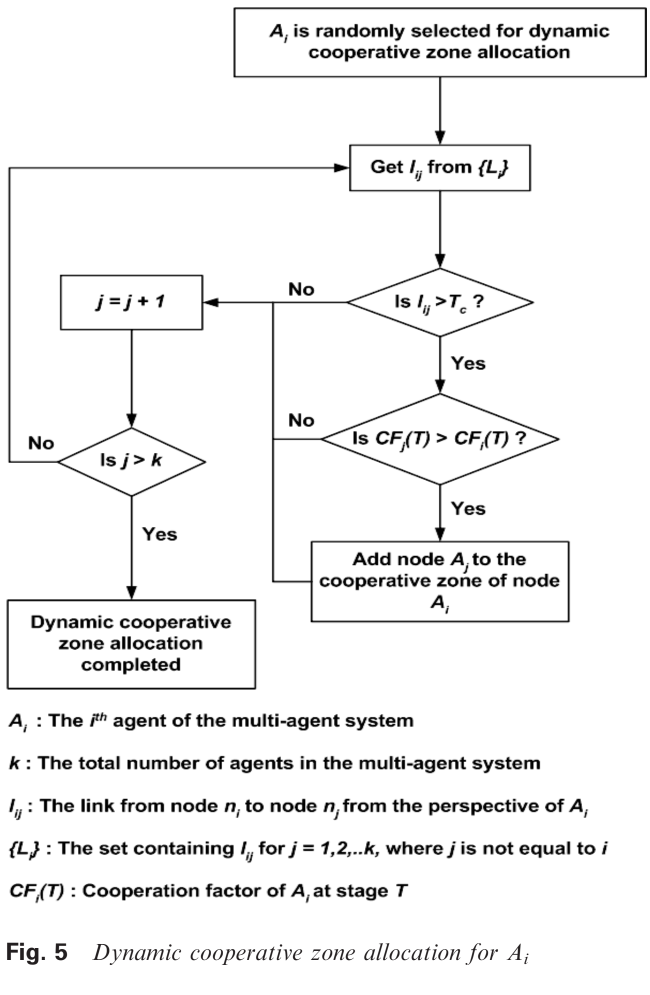

Fig. 5 *Dynamic cooperative zone allocation for A_i*

zone of A_i. Also, A_j will not be further considered in the next round of random selection.

The allocation of the cooperative zone for A_i is complete once every member in the set $\{L_i\}$ has been processed. Hence, in the situation where either one of the following is true for each l_{ij} in the set $\{L_i\}$,

(1) $l_{ij} < T_c$,
(2) $l_{ij} > T_c$ and $CF_j(T) < CF_i(T)$.

The agent A_i will have no agent in its cooperative zone (meaning A_i is isolated). In other words, A_i will only need to consider its own set of $CD_i(T)$ and $CF_i(T)$ as the outputs for node n_i during the time stage T.

Given that the value of each l_{ij} in $\{L_i\}$ is being updated continuously, it can be seen that the cooperative neighbourhood of each A_i will not be static as long as errors are present.

3.4 Enhanced cooperative behaviours among the agents

In this multi-agent system, cooperation is designed to take place within the respective cooperative zones while agents who do not belong to any cooperative zone will not be involved in any collaborative action. Learning and inference-based cooperation is achieved via the integration of cooperation and learning in each agent, which can adjust the size of its cooperative zone depending on how it learns from its previous mistake. In turn, the size of the cooperative zone determines the number of other agents that can affect its future learning process. The cooperative act of sharing of useful information in this single-layer system leads to lateral-based communication in the CE of agents.

3.5 Cooperative parameters update algorithm

The parameters of the various modules are updated in a cooperative manner. An additive decay method is introduced to take into account all the previous errors of an agent, which are in turn decaying iteratively. In this neurobiologically inspired algorithm, previous experiences are stored in each agent in the form of additive errors. A *forgetful mechanism* is implemented in the form of decay functions. The number of errors that an agent receives during each iteration is dependent on the size of its team. The final error is backpropagated to the NN of each agent.

4 Simulation setup

The traffic network used for this paper is similar to the one used in the research work [11]. It is modeled after the Suntec City area, which is a section of the Central Business District (CBD) of Singapore using PARAMICS Modeller (part of the PARAMICS v4 [13] microscopic simulation package). The reason for using PARAMICS is that it is able to accurately model various traffic behaviours such as congestion formation and dispersion, as well as maintain an accurate picture of what is actually happening in the traffic network.

In total, the traffic network contains 330 links and 130 nodes though not all nodes are signalised intersections. Traffic data for coding the traffic network in PARAMICS Modeller is obtained from the Land Transport Authority of Singapore. The total number of signalised intersections in the simulated traffic network that are to be controlled by the agents is 25. As such, 25 agents from the CE as well as the SPSA–NN based multi-agent system [11] are used to provide real-time traffic signal control.

Inductive loop detectors were coded in the simulated traffic network at stop lines of the intersection approaches, similar to the real-world installations. Using the PARAMICS Application Programming Interface (API), information or traffic parameters such as lane occupancy, flow and rate of change of flow is extracted in real-time from the loop detectors. These traffic parameters will then be used as the inputs $I_i(T)$ to the agents at each stage T. Subsequently, the agents generate the traffic signal policies for their respective intersections and these will be implemented into the simulated traffic network using the PARAMICS API. Similar to [11], each agent has eight different types of signal plans in which it can use to control the traffic signals at its intersection. These eight different types of signal plans are designed to cater for different amounts of traffic loading at the intersection and it is up to each agent to choose an appropriate signal plan based on its own perception of the traffic loading of its intersection.

The multi-agent system presented in this paper is coded entirely using Java multi-threading technology. The sampling rate for the agents can be coordinated in order to make sure that the agents make timely responses to the dynamically changing traffic network. For this paper, the agents are tuned to sample the traffic network for the traffic parameters and update the weights of their NNs once every 10 s (simulation time). This essentially is the same sampling rate used by the SPSA–NN based multi-agent system in [11].

Finally, the GLIDE benchmark in [11] is also used for benchmarking in this paper. Note that GLIDE stands for Green Link Determining and it is the local name of SCATS (SCATS stands for Sydney Coordinated Adaptive Traffic System and it is one of the state-of-the-art adaptive traffic signal control systems [13] which is currently used in over 70 urban traffic centres in 15 countries worldwide).

5 Results

In this paper, two performance criteria, 'total mean delay of vehicles' and 'current vehicle mean speed', are used to evaluate the performances of the multi-agent systems as well as the GLIDE benchmark. The microscopic traffic simulation platform of PARAMICS allows detailed measurements of various parameters associated with each vehicle that enters and leaves the traffic network. As such, during the course of the simulation, the delay faced by each vehicle entering and leaving the network was stored in memory and the total mean delay of vehicles was calculated as follows:

$$T_{\mathrm{MD}} = \frac{T_{\mathrm{D}}}{T_{\mathrm{V}}} \tag{1}$$

where, T_{MD} is the total mean delay of vehicles, T_{D} is the total amount of delays faced by all vehicles that entered and left the traffic network during the time when the measurement was taken, and T_{V} is the total number of vehicles that entered and left the traffic network during the time when the measurement was taken. The current vehicle mean speed is the average speed of all the vehicles that are currently in the traffic network. These two performance measures are reflective of the overall traffic condition in the traffic network. For an over-congested traffic network, the total mean delay of the vehicles will be high and the current mean speed of the vehicles that are in the traffic network will be low.

Three simulation scenarios are designed for the evaluation process. They are namely the typical scenario with morning peak (3 h), the short extreme scenario with two peaks (6 h) and long extreme scenario with multiple peaks (24 h). The typical scenario with a single morning peak is used to test the response for the multi-agent systems when they are being implemented to provide real-time traffic signal control of a large traffic network for a short duration of time. The short extreme scenario is designed to increase the level of difficulty of a short simulation run by having two peak periods closely packed together, while the hypothetical 24 h scenario with multiple peaks is used to test the performance of the control systems under very extreme conditions.

5.1 Results for the typical scenario with morning peak (3 h)

For the typical scenario with morning peak (3 h), the two multi-agent systems and the GLIDE benchmark are evaluated by performing 15 separate simulation runs using different random seeds for each of the control systems. The variances of the outcomes of the simulation are small. Hence, the average values are taken to be reasonable representation of a typical outcome. The performance of each technique measured using the respective performance measures is shown in Figs 6 and 7. Note that in these figures, a grey shaded region denotes a peak period.

The SPSA–NN based multi-agent system and the CE slightly outperform the GLIDE benchmark in terms of total mean delay of vehicles. Compared to the GLIDE benchmark, the SPSA–NN based multi-agent system has achieved 15.2% reduction in total mean delay of vehicles at the end of the 3 h simulation, while the CE has achieved an 8.4% reduction. The CE improves the current mean speed of vehicles by ∼13.3% when compared to the GLIDE benchmark. It should be noted that the current mean speed of vehicles for all three techniques increases steadily shortly after the end of the first peak period.

5.2 Results for the short extreme scenario with two peaks (6 h)

The performance of the two multi-agent systems as well as the GLIDE benchmark was further evaluated using a more difficult simulation scenario. As such, the 6 h short extreme scenario with two peak periods was designed, and as in the previous case, 15 separate simulation runs using different random seeds were

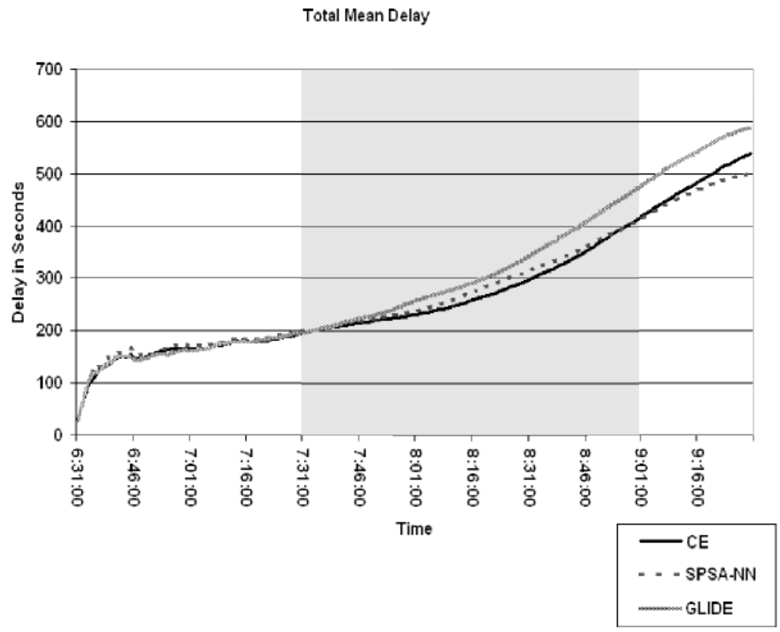

Fig. 6 *Total mean delay for 3 h simulation*

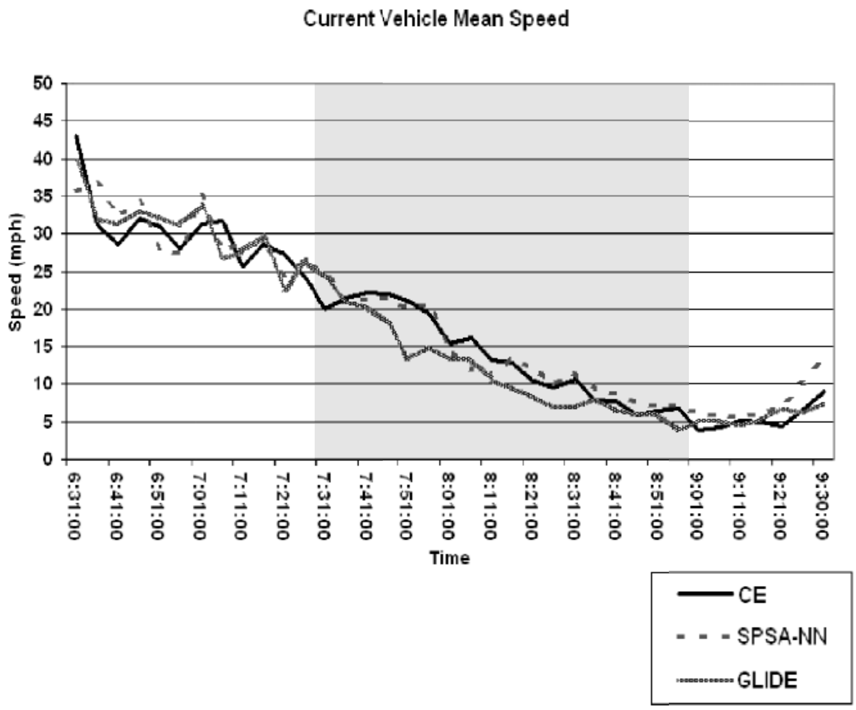

Fig. 7 *Current vehicle mean speed for 3 h simulation*

carried out. Figs 8 and 9 show the performance of the two multi-agent systems and the GLIDE benchmark as evaluated using the two performance measures.

It can be observed that CE-based multi-agent system performs better than SPSA–NN and GLIDE benchmark. Compared to the GLIDE benchmark, the CE reduces the total mean delay by ~47% and the SPSA–NN-based multi-agent system yields a 38% reduction. Fig. 9 shows that the current mean speed of vehicles recovers at the fastest rate at the end of the second peak period when the CE is implemented into the traffic network.

5.3 Long extreme scenario with multiple peaks (24 h)

The long extreme scenario with multiple peaks is a hypothetical scenario to test the robustness of the different techniques under extreme conditions. Fig. 10 shows the number of vehicles over a 24 h period for this extreme scenario. Six separate runs were carried out using different random seeds for each of the three control techniques.

The results for this extreme scenario are given in Tables 1 and 2, summarising the respective average values of the total mean delay and the vehicle mean

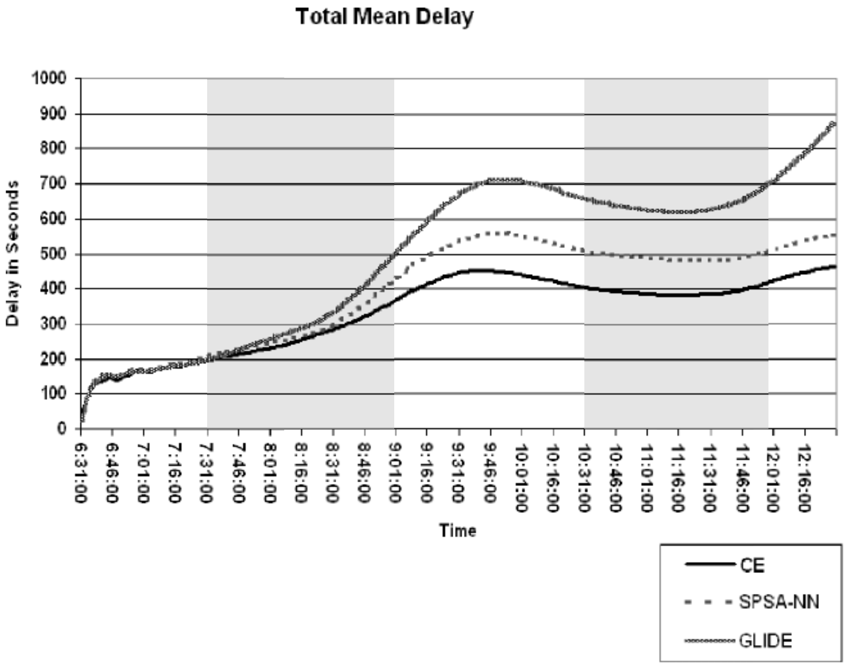

Fig. 8 *Total mean delay for 6 h simulation with two peaks*

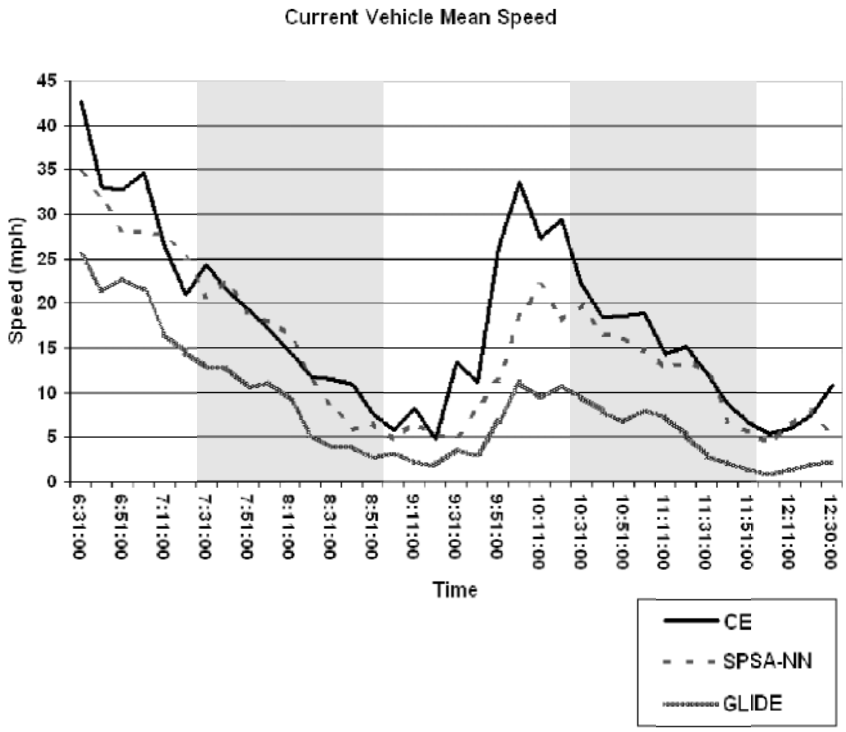

Fig. 9 *Current vehicle mean speed for 6 h simulation with two peaks*

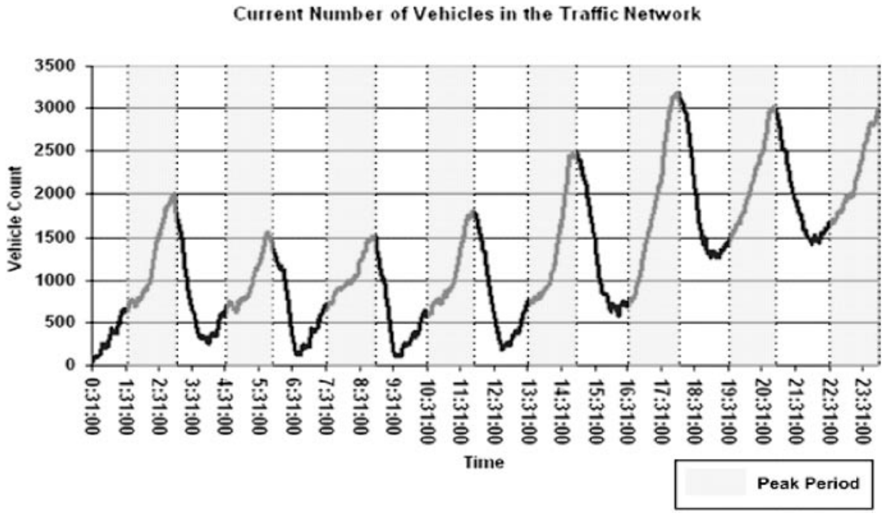

Fig. 10 *Number of vehicles for an extreme scenario with multiple peaks (24 h)*

Table 1: Total mean delay for the long extreme scenario with multiple peaks

	Total mean delay (seconds per vehicle)							
Control technique	1st Peak period	2nd Peak period	3rd Peak period	4th Peak period	5th Peak period	6th Peak period	7th Peak period	8th Peak period
CE	600	750	750	750	750	750	750	800
SPSA-NN	600	750	600	550	500	500	500	850
GLIDE	400	500	600	650	800	1500	2300	3200

Table 2: Current vehicle mean speed for the long extreme scenario with multiple peaks

	Current vehicle mean speed (mph)							
Control technique	1st Peak period	2nd Peak period	3rd Peak period	4th Peak period	5th Peak period	6th Peak period	7th Peak period	8th Peak period
CE	5	5	5	5	2.5	5	4	4
SPSA-NN	5	10	15	7.5	4	5	4	0
GLIDE	7	5	5	5	0	0	0	0

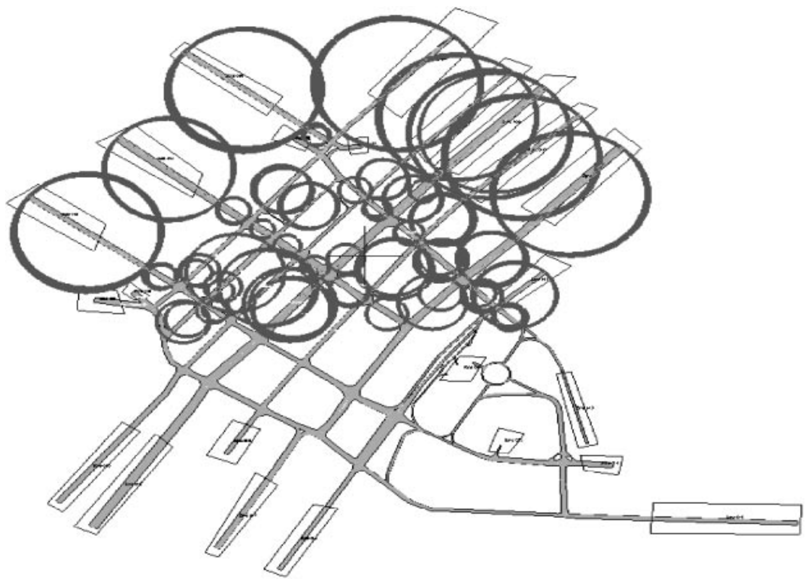

Fig. 11 *Traffic network controlled by SPSA–NN based Multi-agent System after 24 h (long extreme scenario with multiple peaks)*

speed at the end of each peak period. As shown, CE outperforms the other techniques by maintaining the average value of the total mean delay as well as the current vehicle mean speed at a reasonable level throughout the duration of the simulations, thereby demonstrating its applicability in providing effective real-time traffic signal control even under this long extreme scenario.

Under this long extreme scenario with multiple peak periods, the CE manages to maintain positive level of traffic flow even though the level of traffic congestion is rather high.

In order to better illustrate the conditions of the traffic network at the end of the long extreme scenario, two-dimensional top–down screenshots of the traffic network are taken from PARAMICS Modeller in Figs 11–13. These screenshots are captured at 00:30 h (at the end of the 24 h for the extreme scenario). The PARAMICS modelling environment was preset to denote 13 stopped or queued vehicles with a hotspot or red circle. As can be seen from Figs 11 and 12, the traffic network evolves into a pathological state with over saturation at 00:30 h using the SPSA–NN-based controllers and the GLIDE benchmark. The number of hotspots for the traffic network controlled by the GLIDE benchmark is the highest compared to that of the traffic network controlled by the multi-agent systems. This is likely to be the result of the steady increase of total mean delay and the drop in the current vehicle mean speed to zero as early as after the fifth peak period when the GLIDE benchmark is implemented. On the other hand, using the proposed CE-based multi-agent system, congestions are confined to the central region of the traffic network. The number of congested

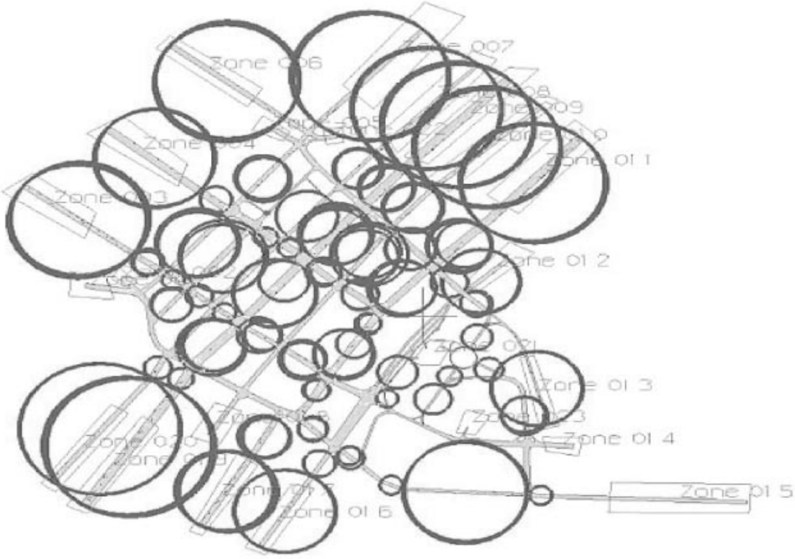

Fig. 12 *Traffic network controlled by the GLIDE benchmark after 24 h (long extreme scenario with multiple peaks)*

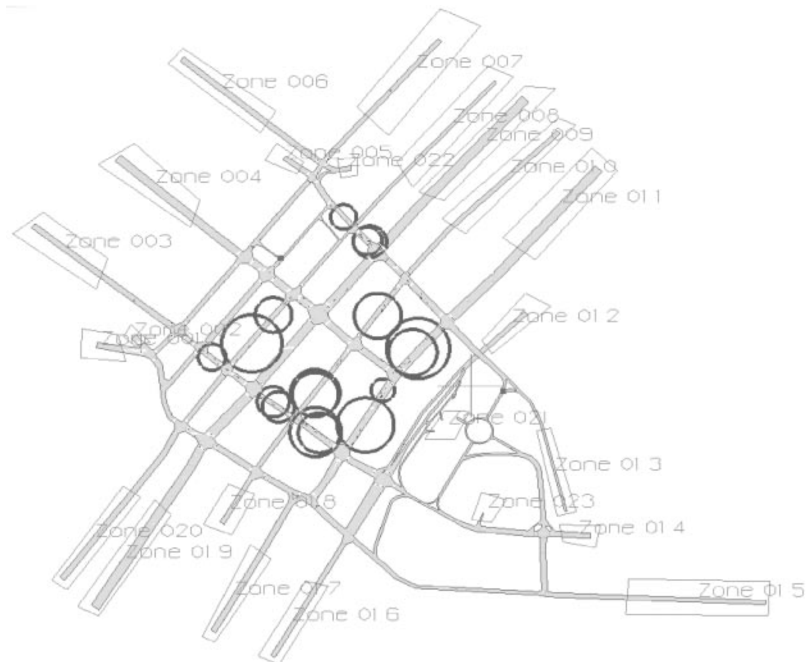

Fig. 13 *Traffic network controlled by the CE after 24 h (long extreme scenario with multiple peaks)*

links is reduced to less than ten as against more than 30 for GLIDE and SPSA–NN.

8 Conclusion

A new CE of agents has been presented in this paper. Enhanced cooperative behaviours such as cooperation within dynamic cooperative zones and dynamic lateral communication for sharing of useful information among agents are implemented in this multi-agent system. Using these enhanced cooperative behaviours as a basis, a stochastic cooperative parameter update algorithm has been designed to improve the online learning and update process for each agent. The performance of the proposed approach has been compared with the SPSA–NN based multi-agent system as well as the GLIDE benchmark via a thorough evaluation process. Empirical results have shown that the performance of the CE is superior compared to the other two techniques under extreme 6 and 24 h simulation scenarios. This implies that the advanced cooperative features in the CE have the desired positive effect on the performance of the agents. The promising results obtained in this study pave way for further applications of similar cooperative features in multi-agent systems.

9 References

1 Saito, M., and Fan, J.: 'Artificial neural network-based heuristic optimal traffic signal timing', *Comput.-Aided Civ. Infrastruct. Eng.*, 2000, pp. 281–291

2 Lee, J.H., and Lee-Kwang, H.: 'Distributed and cooperative fuzzy controllers for traffic intersections group', *IEEE Trans. Syst. Man Cybern. C, Appl. Rev.*, 1999, **29**, (2)

3 Niitymaki, J.: 'Using fuzzy logic to control traffic signals at multiphase intersections'. Proc. Dortmunder Fuzzy-Tage, Dortmund, Germany, 1999. Computational Intelligence: Theory and Applications, pp. 354–362

4 Trabia, M.B., Kaseko, M.S., and Ande, M.: 'A two-stage fuzzy logic controller for traffic signals', *Transp. Res. C, Emerg. Technol.*, 1999, pp. 353–367

5 Niitymaki, J.: 'General fuzzy rule base for isolated traffic signal control-rule formation', *Transp. Plan. Technol.*, 2001, pp. 237–247

6 Chiu, S., and Chand, S.: 'Self-organising traffic control via fuzzy logic'. Proc. 32nd IEEE Conf. Decision Control, 1993, pp. 1987–1902

7 Nakamiti, G., and Gomide, F.: 'Fuzzy sets in distributed traffic control'. Proc. 5th IEEE Int. Conf. Fuzzy Systems, 1996, pp. 1617–1623

8 Mikami, S., and Kakazu, Y.: 'Genetic reinforcement learning for cooperative traffic signal control'. Proc. 1st IEEE Conf. Evolutionary Computation, 1994, vol. 1 pp. 223–228

9 Manikonda, V., Levy, R., Satapathy, G., Lovell, J.D., Chang, C.P., and Teittinen, A.: 'Autonomous agents for traffic simulation and control'. Proc. 80th Meeting National Research Council (U.S.), Transportation Research Board, Washington, DC 2001

10 Takahashi, S., Nakamura, S., Kazama, H., and Fujikura, H. 'Adaptive traffic signal control for the fluctuations of the flow using a genetic algorithm'. Proc. 8th Int. Conf. Urban Transport and the Environment in the 21st Century, 2002, pp. 239–247

11 Choy, M.C., Srinivasan, D., and Cheu, R.L.: 'Simultaneous perturbation stochastic approximation based neural networks for online learning'. Proc. 7th IEEE Int. Conf. Intelligent Transportation Systems, October 2004, pp. 1038–1044

12 Spall, J.C., and Chin, D.C.: 'Traffic-responsive signal timing for system-wide traffic control', *Transp. Res. C*, 1997, **5**, (3/4), pp. 153–163

13 Quadstone: 'PARAMICS Modeller v4.0 User Guide and Reference Manual'. (Quadstone Ltd, Edinburgh, UK, 2002)

14 Wolshon, P.B., and Taylor, W.C.: 'Analysis of intersection delay under real-time adaptive signal control', *Transp. Res. C, Emerg. Technol.*, 1999, **7C**, (1), pp. 53–72