World Applied Sciences Journal 5 (5): 525-530, 2008
ISSN 1818-4952

Coordination between Traffic Signals Based on Cooperative

[1]*S.S. Shamshirband,* [2]*H. Shirgahi,* [3]*M.Gholami and* [4]*B. Kia*

[1]Department of Computer, Islamic Azad University, Chalous Branch, Chalous, Iran
[2]Department of Computer, Islamic Azad University, Jouybar Branch, Jouybar, Iran
[3]Iranian Academic Center for Education, Culture and Research Mazandaran Branch, Sari, Iran
[4]Department of Science, Islamic Azad University, Chalous Branch, Chalous, Iran

Abstract: The single traffic signal control agent improves its control ability with the Multiagents-learning method. This paper proposes a new cooperative learning method; called weighted strategy sharing (WSS) is presented. In this method, each agent measures the expertness of its teammates and assigns a weight to their knowledge and learns from them accordingly. The presented methods are tested on three traffic lights. Also, the effect of the communication noise, as a source of uncertainty, on the cooperative learning method is studied. Moreover, the Qtable of one of the cooperative agents is changed randomly and its effects on the presented methods are examined. Results using cooperative traffic agents are compared to results of control simulations where non-cooperative agents were deployed. The result indicates that the new coordination method proposed in this paper is effective.

Key words: Multiagents . cooperative learning . expertness . qlearning

INTRODUCTION

The increase in urbanisation and traffic congestion create an urgent need to operate our transportation systems with maximum efficiency. Realtime traffic signal control is an integral part of modern Urban Traffic Control Systems aimed at achieving optimal Utilization of the road network. Providing effective real time traffic signal control for a large comp lex 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. Optimization 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 [1-6].

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. 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.

Intelligent software agent is an autonomous computer program, which interacts with and assists an end user in certain computer related tasks [1]. In any agent, there is always a certain level of intelligence. The level of the Intelligence could vary from pre-determined roles and responsibilities to a learning entity. Multi-Agent System is the aggregate of agents, whose object is to decompose the large system to several small systems which communicate and coordinate with each other and can he extended easily.

Agent-based simulations are models where multiple entities sense and stochastically respond to conditions in their local environments, mimicking complex large-scale system behavior [2]. The urban traffic system is a much complex system, which involved many entities and the relationship among them are Complicated. Therefore, the Application of MAS into the simulation of traffic system is suitable and efficient [3]. One of the

Corresponding Author: Dr. S.S. Shamshirband, Department of Computer, Islamic Azad University, Chalous Branch, Chalous, Iran

most important issues for a learner agent is the assessment of the behavior and the intelligence level of the other agents. In addition, the learner agent must assign a relative weight to the other agents' knowledge and use it accordingly. In general, these three issues are very complex and need careful attention. Therefore, in this paper attention has been paid to find some solutions for homogeneous, independent and cooperative Q-learning agents. In some studies a new cooperative learning strategy, called weighted strategy sharing (WSS) and some expertness measuring methods are introduced [3, 4].

Some studies assumed that the learner agents cooperate only with the more expert agents. Also assumed that, the communication is perfect and all of the agents are reliable, therefore it is considered that all of the agents could learn from each other.in addition, effects of the communication noise as a source of uncertainty on the cooperative learning are studied. Moreover, the Q-table of one of the cooperative agents is changed randomly and its effects on the presented method are examined [5, 6].

In this paper, some kind of traffic signal control agents is developed in the agent-based simulation environment and the coordination strategy between the control agents is introduced in detail. In the next section Then, WSS is briefly introduced and some expertness measures are presented. Section 3 introduces the detail related with the coordination between more than two traffic control agents. In sections 4, the effectiveness of the coordination strategy is proved in the simulation system. Finally, the conclusion is of this paper is given is section 5.

TRAFFIC SIGNAL CONTROL AGENT (TSCA)

According to the difference of the control scope, there are there methods for the realization of the traffic light control agent:

- Every agent controls only a phase of an intersection [4-7]. In this situation, when there are many intersections in the road network, the amount of the agents is too large. And as a result, the communication and the coordination between the agents is much complex.
- Every agent controls all the phases of an intersection [5-7]. The control agent of this kind could coordinate the benefit of all the phases of an intersection. The coordination between different intersections depends on the social rules and the game theory.

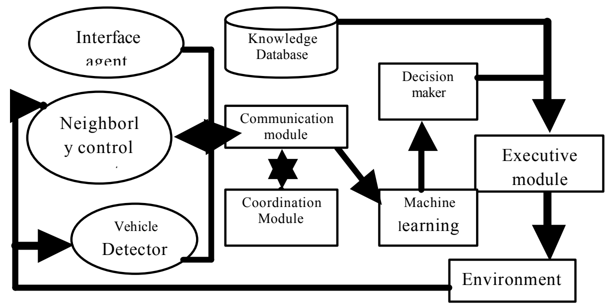

Fig. 1: Model of the control agent

- Every agent controls an area of intersections [8]. The separation of the area should be done firstly and then, it's hard to change. The shortcoming of this method is that it isn't flexible. We design our control agent on the base of method (2). The model of it is shown in Fig. 1.

The process of the control is as follows: first, the vehicle detector and the neighborly control agents send the information to the agent; then, it makes the decision based on the received information and the knowledge it owns; finally the decision is put into control action by the Executive module.

In RL, an agent tries to maximize a scalar evaluation (Reward or punishment) of its interaction with the environment. The goal of a RL system is to find an optimal policy which maps the state of the environment to an action which in turn will maximize the accumulated future rewards. Most RL techniques are based on Finite Markov Decision Processes (FMDP) causing finite state and action spaces. The main advantage of RL is that it does not use any knowledge database, as do most forms of machine learning, making this class of learning suitable for online learning. The main disadvantages are a longer convergence time and the lack of generalization among continuous variables. The latter is one of the most active research topics in RL [9-13].

The control actions of the traffic light control agent are: 'Extend' or 'terminate'. 'Extend' means to "extend the original lamp state to the next time interval"; 'Terminate' means to "change the lamp state". We suppose that the states of the lamp are only green and red, the yellow state is eliminated.

In this paper, the reward of the control agent is fuzzy reward determines whether to extend or terminate the current green Phase based on a set of fuzzy rules.

QC = Average queue length on the lanes served by the current green, in veh/lane.

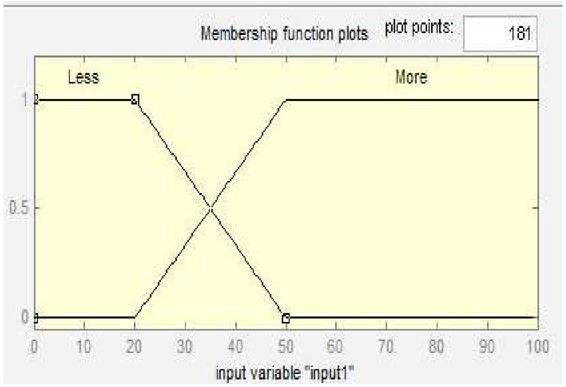

Fig. 2: Fuzzy set for traffic flow

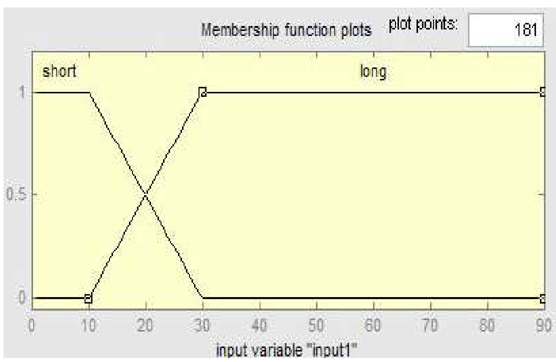

Fig. 3: Fuzzy set for delay time

QN = Average queue length on lanes with red which may receive green in The next phase, in veh/lane.
AR = Average arrival rate on lanes with the current green, in veh/sec/lan.

The decision making process based on a set of fuzzy rules which takes into account the traffic Conditions with the current and next phases. The general format of the fuzzy rules is as follows:

If {QC is X1} and {AR is X2}
and {QN is X3} Then {E or T}.

where, X1, X2, X3 = natural language expressions of traffic conditions of respective variables.

The Q-Value is a function of the main factors influencing the control strategy, which include the traffic flow of the green phase (AR); the number of the waiting vehicles in red phase (QN); Average queue length on the lanes served by the current green, in veh/lane (QC).

Then, the Q-Value can be determined by the following function:

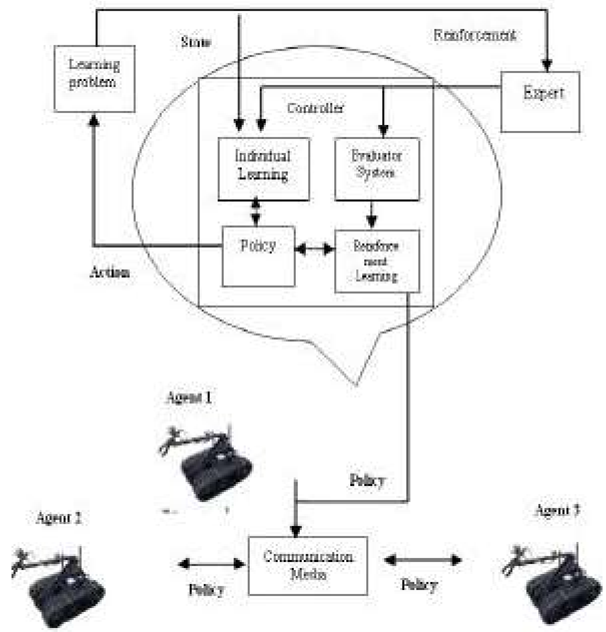

Fig. 4: Weighted strategy sharing

$$\hat{Q} = f((AR, QN, QC), a, \theta) \qquad (1)$$

where, (AR, QN, QC) is the input state, a is the chosen action, θ is the weight vector of the neural network.

The possibility of choosing action a is determined by the following function:

$$P_a = \frac{e^{Q(a)/\tau}}{\sum_{b=1}^{n} e^{Q(b)/\tau}} \qquad (2)$$

Where, n is the number of the actions; Q (a) is the evaluation value of action a; τ is a positive number named as temperature. The higher the temperature, the more average every action is selected.

COORDINATION MECHANISM

WSS method: In the WSS method [16] (Fig. 3), it is assumed that *n* homogeneous one-step Q-learning agents learn in some distinct environments and no hidden state is produced [10-17].

The agents learn in two modes: individual learning mode and cooperative learning mode (Fig. 5). At first, all of the agents are in individual learning mode. Agent i executes ti learning trials. Each learning trial starts from a random state and ends when the agent reaches the goal. After a specified number of individual trials, all agents switch to cooperative learning mode.

Initialize
While not end of learning do
Begin
If ← in individual learning mode then
Begin individual learning
X_i ← Find Current State()
a_i ← Select Action(a_i)
Do Action (a_i)
r_i ← Get Reward ()
y_i ← Go To Next State ()
$V(y_i) \leftarrow Max_{b \in actions} Q(y_i, b)$
$Q_i^{new}(x_i, a_i) := (1-\beta_i) Q_i^{old}(x_i, a_i) + \beta_i (r_i + \gamma_i V(y_i))$
e_i ← Update Expertness (r_i)
End
Else Cooperative Learning
Begin
For j: = 1 to n do
e_i ← Get Expertness (A_j)
$Q_i^{new} \leftarrow 0$
For j: = 1 to n do
Begin
W_{ij} ← Compute Weights (i, j, $e_1 \ldots e_n$)
$Q_j^{old} \leftarrow GetQ(A_i)$
$Q_i^{new} \leftarrow Q_i^{new} + W_{ij} * Q_j^{old}$
End

Fig. 5: Algorithm, weighted sharing algorithm for agent (Ai)

In cooperative learning mode, each learning agent assigns some weights to the other agents' Q-tables with respect to their relative expertness. Then, each agent takes the weighted average of the others' Q-tables and uses the resulted table as its new Q-table.

$$Q_i^{new} \leftarrow \sum_{j=1}^{n} (W_{ij} \times Q_j^{old}) \tag{3}$$

Expertness criteria: In the WSS method, Wij is a measure of agent reliance on the knowledge and the experiences of agent. Here we argue that this weight is a function of the agents' relative expertness. In the strategy sharing method, expertness of the agents are assumed to be equal. Some studies used the user judgment for specifying the expert agent. This method requires continuous human supervision.However, some studies specified the expert agents by means of their successes and failures during current moves and considered the expertness criterion as an algebraic sum of the reinforcement signals in that time interval. This means that more successes and fewer failures are considered a sign of a higher degree of expertness. This expertness measuring method is not Optimal in some situations. For example, the agent that has faced many failures has some useful knowledge to be learned from it. In other words, it is possible that this agent does not know the ways arriving at the goal, but it is aware of those not leading to its target and can avoid them. Also, an agent at the beginning of its learning process is fewer experts than those learned for a longer time and naturally has confronted more failures. Considering the discussions, one expertness measure is introduced. These measures include the following [15-17].

A) Absolute (Abs): A sum of the absolute value of the reinforcement signals

$$e_i^{Abs} = \sum_{t=1}^{now} |r_i(t)| \tag{4}$$

Abs considers both rewards and punishments as a sign of being experienced.

Type the title approximately 2.5 centimeters (1 inch) below the first line of the page and use 20 points type-font size in bold. Center the title (horizontally) on the page. Leave approximately 1 centimeter (0.4-inches) between the title and the name and address of yourself (and of your co-authors, if any.) Type name(s) and address(s) in 11 points and center them (horizontally) on the page. Note that authors are advised not to include their email addresses (unless they really want to.)

B) Weight assigning mechanisms

Learning from All (LA): It can be said that all agents have some valuable knowledge to be learned. When using all agents' knowledge, the simplest formula to assign weight to agent j knowledge by learner could be

$$W_{ij} = \frac{e_j}{\sum_{k=1}^{n} e_k} \tag{5}$$

where n is the number of the agents and ek is the amount of the expertness of agent k. In this method, effects of agent j knowledge on all learners are equal, i.e.

$$W_{1j} = W_{2j} = \ldots W_{nj}$$

Also all of Q-tables become homogeneous after each cooperation step.

IMPLEMENTATION

We have constructed a prototype traffic simulator program to test the efficiency of the coordination mechanism we proposed. The programming language we used to build the simulator is VC#.Net.

The prototype of simulator: The simulator prototype is programmed mainly to verify the efficiency of the coordination mechanism we proposed in this paper. The traffic environment includes: 2-lane roads, 3 intersections, traffic light control agent and vehicles. The main reason we choose only 3 intersections is that the computational complexity of more than 3 intersections is too high and the work of this paper is just an exploration. Further study should be done in the future to simulate the Coordination among more than three intersections.

RESULT AND DISCUSSION

The road network in the simulator is shown in Fig. 6.

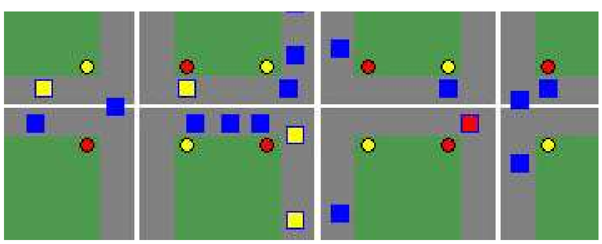

Fig. 5: The road network

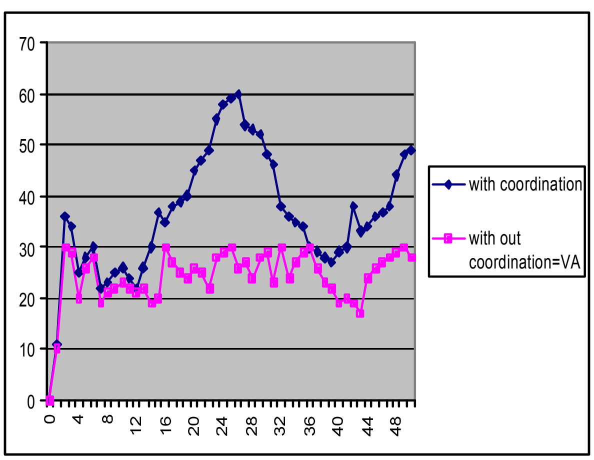

Fig. 6: Compare between result of with and without coordination

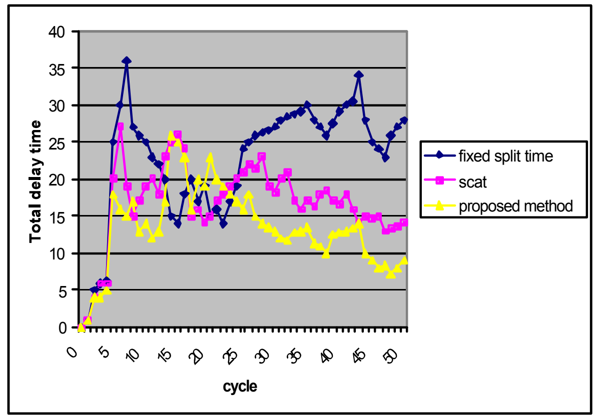

Fig. 7: Compare proposed method with scat and fix time split

In this study supposed that there are only two phases in the three intersections. The percent of the vehicles turning left is 0.2. Inputs of car default are 40 vehicles.

Figure 6 and 7 shows the result of the simulation. From Fig. 6 we can see that the coordination mechanism proposed in this paper is efficient, especially when the traffic flow of horizontal direction is much more than the vertical direction.

CONCLUSION

In this paper, one weight-assigning procedure for the Weighted Strategy Sharing (WSS) methods was introduced. Also, some criteria to measure the expertness of the agents were presented. The introduced methods were tested on the Traffic Lights problem. Detection of the agents with incorrect knowledge and minimizing their effects on the cooperative group learning is another Direction for future research. To make the mechanism suitable for more intersections, the algorithm should be optimized to reduce the learning time of the TSCAs. The simulator prototype of this paper is only a primary system. To be a more complete and universal traffic simulator, many of the elements should be improved in the future work.

REFERENCES

1. Gary, S., H. Tan and K.L. Hui, 1998. Applying intelligent agent technology as the platform for simulation. Proceedings of the Simulation Symposium, pp: 180-187.

2. Sanchez, S.M. and T.W. Lucas, 1972. Exploring the world of agent based simulations: simple models, complex analyses, Proceedings of the Simulation Conference, pp.116-126.
3. Zhongzhi, S., 1998. The Advanced Artificial Intelligence. Beijing: Science Press, Chapter 10, pp: 223-226.
4. Pan, G.C. and B. Maddox, 1995. A Framework for Distributed Reinforcement Learning, Lecture Notes in Artificial Intelligence, Adaptation and Learning in Multi-Agent Systems, pp: 97-112.
5. Liu, Z., 2007. A Survey of Intelligence Methods in Urban Traffic Signal Control, International Journal of Computer Science and Network Security, 7(7): 105-112.
6. Lucia, A. and C. Bazan, 1994. Traffic Signal Coordination Based on Distributed Problem Solving. 7' Fanfares Symposium on Transportation Systems: Theory and Application of Advanced Technology, Tianjin, China, pp: 957-962.
7. Ming, L.X. and F.Y. Wang, 2001. Study of city area traffic coordination control on the basis of agent. F'roceedings of the IEEE intelligent Transportation Systems Conference, Singapore, September, pp: 758-761.
8. Ossowaki, S., J. Cuena and A. Garcia, 1998. A Case of Multiagent Decision Support: Using autonomous agents for urban traffic control. Lecture Notes in Artificial Intelligence, 1484: 100-111.
9. Chen, W. and K. Decker, 2004. Developing Altcmative Mechanisms for Multiagent Coordination. Lecture Notes in Computer Science, Springer-Verlag Heidelberg, 2413: 63-76.
10. Sutton, R.S., 1988. Learning to predict by the methods of temporal differences. Mach. Learn, 3: 9-44.
11. Sutton, R.S., 1998. Machine Learning: Special Issue on Reinforcement Learning. Cambridge, MA: MIT Press, 8: 3-4.
12. Yamaguchi, T., Y. Tanaka and M. Yachida, 1997. Speed up reinforcement learning between two agents with adaptive mimetism. Proc. IEEE Conf. Intl. Robot. Syst. (IROS), pp: 594-600.
13. Friedrich, H., M. Kaiser, O. Ragalla and R. Dillmann, 1996. Learning and communication in multi-agent systems. Distributed Artificial Intelligence Meets Machine Learning. Weiss, G. (Ed.). New York: Springer-Verlag, 1221: 259-275.
14. Watkins, C.J., 1989. Learning from delayed rewards. Ph.D. dissertation, King's College, Cambridge, UK.
15. Watkins, C.J. and P. Dayan, 1998. Q-learning (technical note). Machine Learning: Special Issue on Reinforcement Learning. Cambridge, MA: MIT Press, pp: 55-68.
16. Tan, M., 1993. Multi-agent reinforcement learning: Independent vs. cooperative agents. In: Proc. 10th Intl. Conf. Machine Learning, Amherst, MA.
17. Friedrich, H., 1996. Learning and communication in multi-agent systems, in Distributed Artificial Intelligence Meets Machine Learning, Weiss, G., (Ed.). New York: Springer-Verlag, 1221: 259-275.