A Distributed Approach for Coordination Between Traffic Lights Based on Game Theory

Shahaboddin Shamshirband
Department of Computer System and Technology, University of Malaya (UM), Malaysia

Abstract: *Traffic signal control agent can improve its control ability by using the NNQ-learning method. This paper proposes a neural network Q-learning approach with fuzzy reward designed for online learning of traffic lights behaviors. The Q-function table usually becomes too large for the required state/action resolution. In these cases, tabular Q-learning needs a very long learning time and memory requirements which makes the implementation of the algorithm impractical, in real-time control architecture. We considered the problem of coordinating three traffic signals control. The coordination is done using control agents and the concept of game theory. To test the efficiency of the coordination mechanism, a prototype traffic simulator was programmed in visual Studion.net. Results using cooperative traffic agents are compared to results of control simulations where non-cooperative agents were deployed. It indicated that the new coordination method proposed in this paper is effective.*

Keywords: *Multiagents, NNQ-learning, fuzzy reward, coordination, cooperative, game theory.*

Received May 24, 2009; accepted January 3, 2010

1. Introduction

The increase in urbanization and, hence, 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 utilization 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.

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. 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 [12]. 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 objective is to decompose a large system to several small systems which communicate and coordinate with each other and can be 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 [17].

The urban traffic system is a very complex system, which involves many entities and the relationship among them are complicated. Therefore, the application of Multi-Agent System (MAS) into the simulation of traffic system is suitable and efficient [13]. According to their ability, the agents can be classified into three types: Reactive, cognitive, and hybrid. Reactive agents make their decisions usually based on a very limited amount of information, and simple situation-action rules. Cognitive agents maintain an internal representation of their world and there is an explicit mental state that can he modified, by some form of symbolic reasoning [5]. In some situations, both reactive and cognitive ability are necessary, such kind of agent is hybrid agent. As a complex system, differentiation of agents could be used in the simulator of urban traffic system [2].

In this paper, a traffic signal control agent is developed within the agent-based simulation environment. Also, the coordination strategy between the control agents is introduced in detail. The model of the control agent based on Neural Network Q-learning NNQ is described in section 2 and its replacement with tabular Q-learning. Section 3 introduces the details concerning the coordination between more than two traffic control agents. In section 4, the effectiveness of the coordination strategy is proved in the simulation system. Finally, the conclusion of this paper is given is section 5.

2. Traffic Signal Control Agent

Based on the properties of the control scope, there are methods for the realization of the traffic light control agent:

1. Every agent controls only a phase of an intersection [15]. In this situation, when there are many intersections in the road network, the number of agents is too large. Therefore, the communication and the coordination between agents is very complex.
2. Every agent controls all the phases of one intersection only [1, 6, 12, 16]. The control agent of this kind thus coordinates all the phases of one intersection independent of other agents. The coordination between different nearby intersections depends on the social rules and the concept of game theory.
3. Every agent controls an area of intersections [13], the separation of the area should be done firstly, and then, it is usually fixed and difficult to change. The shortcoming of this method is that it is not flexible.

We have designed our control agent on the base of method 2. The overall general model of this type of agent is presented in [15]. The process of the control is as follows: First, the vehicle detector and all the neighboring control agents send the information to the agent that is to make a decision. The decision is than made based on the information just received and the knowledge the agent owns. Finally, the decision is put into control action by the executive module.

In Reinforcement Learning (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. It makes 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 [14].

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, thus, eliminated. In our paper, the reward of the control agent is a fuzzy one which determines whether to extend or terminate the current green phase based on a set of fuzzy rules. To formulate the problem, the following notations are used:

- *QC = Average queue length on the lanes served by the current green, in veh/lane.*
- *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 is based on a set of fuzzy rules which takes into account the traffic conditions in 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*, and *X3* are natural language expressions of traffic conditions of respective variables. Figures 1 and 2 are graphic representations for fuzzy sets less, more, short, and long.

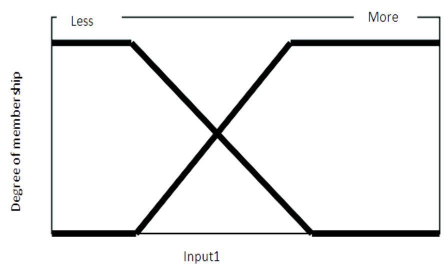

Figure 1. Fuzzy set for traffic flow.

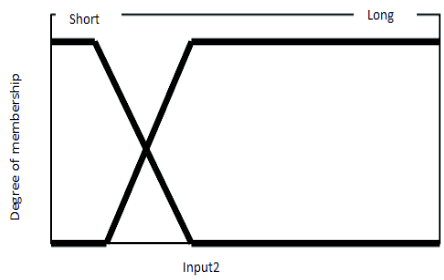

Figure 2. Fuzzy set for delay.

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 waiting vehicles in the red phase QN, the 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:

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

Where, AR, QN, QC is the input state, a is the chosen action, and θ is the weight vector of the neural network. The possibility of choosing action a is determined by the following function:

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

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

3. Neural Q-learning

This paper proposes a NNQL approach designed for online learning of traffic lights control. However, when working with continuous states and actions, as is usual for agents, the Q-function table becomes too large for the required state/action resolution. In these cases, tabular Q-learning needs a very long learning time and memory requirements which, in turn, makes the implementation of the real-time control algorithm impractical. The use of a Neural Network (NN) technique in order to deal simultaneously deal with states and actions reduces the number of values stored in the Q-function table to a set of NN weights. The implementation of a feed-forward NN with the back propagation algorithm [11] is known as direct Q-learning [3]. Residual Algorithms: Reinforcement learning with unction approximation [15].

Direct Q-learning algorithm has no convergence proof and turned out to be unstable when trying to learn a behavior. The instability is caused by the lack of weight updating in the overall state/action space. The optimal Q-function is only learnt in the current state zone. That is, the Q-values learnt in past states is not maintained and therefore had to be learnt each time, thus, causing the instability. To solve this limitation the proposed Neural-Q-learning based behaviors maintains a database of the most recent learning samples. All the samples of this database are used at each iteration to update the weights of the NN. This assures a generalization in the whole visited state/action space instead of a local generalization in the current visited space. Each learning sample is composed of the initial state st, the action at, the new state st+1 and the reward rt. The structure and phases of the proposed neural Q-learning algorithm is shown in Figure 3. The Q-function approximated by the NN is:

$$Q^\wedge(s, a) = rt + ymaxQn(nextstate\ allocation) \tag{3}$$

The inputs are the continuous state and actions, and the output is the Q-value. Based on the output value, the error is found and the weights are updated using the standard back propagation algorithm. A two layer NN has been used with a hyperbolic tangent and lineal activation functions for the first and second layers, respectively. Weights are initialized randomly. To find the action which maximizes the Q-value, the network evaluates all the possible actions which could be applied. Although actions are continuous, a finite set, which guarantees sufficient resolution, is used. To maintain the real time execution of the behavior, two execution threads are used. The thread with higher priority is in charge of acquiring new learning samples and generating control actions at the frequency of the behavior-based control system. Phases 1, 2 and 4 are computed by this thread, see Figure 3. The second thread is used to continually update the weights of the NN. This is represented by phase 3.

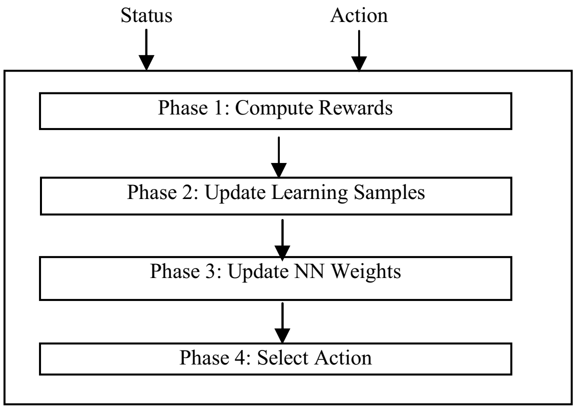

Figure 3. Neural q-learning structure.

4. Coordination Mechanism

Coordination, which is the process that an agent reasons about its local actions and the (anticipated) actions of others to try to ensure a reasonable social act in a coherent fashion, is an important issue in multi-agent systems [4].

Coordination is a complicated process that typically consists of several operations: Exchanging local information, detecting interactions, deciding whether or not to coordinate, proposing, analyzing, refining and forming commitments, sharing results, and so on. The coordination between the agents can he divided into two aspects: Subjective coordination and objective coordination. The distinction between subjective and objective coordination plays a fundamental role in the engineering of multi-agent systems. In subjective approaches, coordination is the result of the attitudes of each individual towards the organization/society it belongs to. Objective approaches, instead, promote the separation between the individual perception of coordination and the global coordination issues, enabling the modeling and shaping of the interaction space independently of the interacting entities [13].

Generally speaking, there are three ways to coordinate multiage systems: Associate coordination mechanisms with each individual agent, construct a special agent that functions as a centralized coordinator, or the combination of the two methods. All of these solutions have both good and had aspects and many possible realizations in different environments. For the purpose of TSCA, we choose the combined method. The coordination between the TSCAs is mainly depended on the two parties involved, but sometimes it can also receive some instructions from the managing agent. The mechanism of the coordination among the TSCAs is mainly based on the game theory, and the social rules and knowledge.

4.1. Implemntation

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

4.2. The Prototype of the 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. Further study should be done in the future to simulate the coordination among more than three intersections.
The coordination between three TSCAs is based on the non-zero game theory. The game can be described as:

$$\Gamma = (agentA, agentB, agentC\ ;\ A, B, C)$$

Where A and B and C represent the game matrix of agents A, B, respectively. The action set of the TSCAs includes two sets:

E = Extension of green phase, and T = Termination of green phase

The utility value represents the benefits gained by taking different action. In this paper, the utility value is set to the Q-value. The game matrix a matrix with three rows and three columns as follows:

Agent c=E

((Qa(E,E,E,),Qb(E,E,E),Qc(E,E,E)) (Qa(E,T,E),Qb(E,T,E)Qc(E,T,E))
(Qa(T,E,E),Qb(T,E,E),Qc(T,E,E)) (Qa(T,T,T),Qb(T,T,E).Qc(T,T,E)

Agent c= T

((Qa(E,E,E,),Qb(E,E,E),Qc(E,E,E)) (Qa(E,T,E),Qb(E,T,E)Qc(E,T,E))
(Qa(T,E,E),Qb(T,E,E),Qc(T,E,E)) (Qa(T,T,T),Qb(T,T,E).Qc(T,T,E)

Where, *Qa* (E, E, E) represents the utility value of agent A when agent A and agent B and agent C all choose the action of 'EXTEND', *Qa* (E, E ,E) represents the utility value of agent B when agent A and agent B and agent C all choose the action 'EXTEND', the rest may be deduced by analogy.

In this paper, an algorithm is designed to solve the Coordination problem directly, and no negotiation should be used. The algorithm is as follows: in this algorithm, we use super agents for coordination. Each agent plays against super agents. The agent in the middle intersection is the super agent and the other two agents play against the super agent. The game is played as a repeating game:

1. From Q-values utility values is computed and the utility matrix is determined.
2. If the action to be taken exists in the matrix, the TSCAs choose their action accordingly and the coordination process ends, otherwise, execute step 3.
3. Compute the maximum and the minimum utility value of the two TSCAs.
4. Search for the Pareto solution (U, V) and (V, Z).
5. Compute the hybrid strategy of each TSCA based on (U, V), (V, Z) and choose the final action.

The coordination is proposed by the TSCA on the base of its demand, if the receiver doesn't want to coordinate with the invite, it can refuse the request. If, otherwise, it wants to coordinate, they first search to determine if suitable social rules exist, if such rules exist, the rule is applied. Otherwise, the coordination algorithm based on game theory is used.

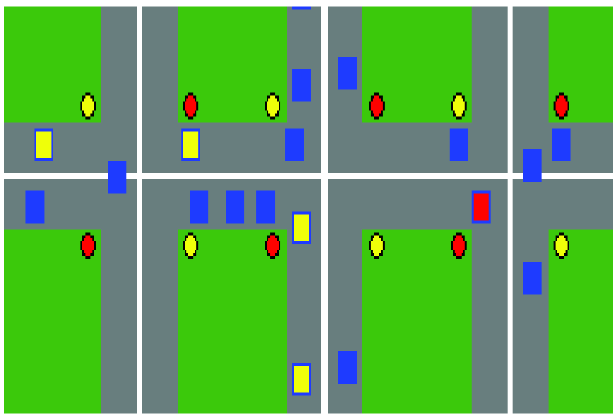

Figure 4. The three intersection road network.

5. Result and Discussion

The road network which is used for our simulation is shown in Figure 4. We assume that there are only two phases, red and green, for every intersection of the three intersections system. The percent of the vehicles turning left is considered to be 0.2. The default number of vehicle input is assumed to be 40. Figures 5 and 6 show the simulation results. It can be seen 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.

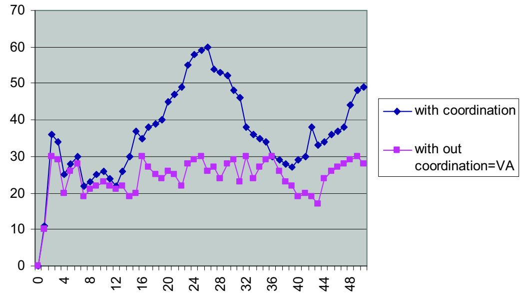

Figure 5. Compare between result of with and without coordination.

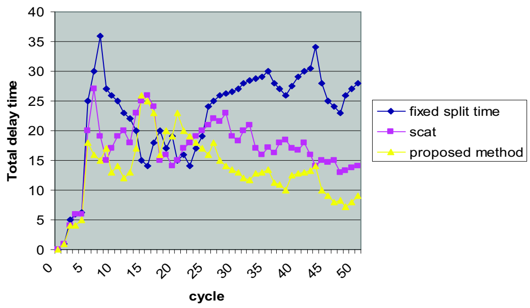

Figure 6. Compare proposed method with scat and fix time split.

6. Conclusions

The study of this paper indicates that the NN Q-learning can be applied into the control of the traffic signal lights, and the coordination mechanism between the three TSCAs in a three-intersection road network is efficient. To make the mechanism suitable for more intersections, the algorithm should be optimized to reduce the learning time of the TSCAs.

The simulator prototype presented in this paper is a primitive one thus; further research is needed to extend it to a practical traffic control system.

References

[1] Baird C., "Residual Algorithms Reinforcement Learning with Function Approximation, Machine Learning," *in Proceedings of the 12th International Conference*, San Francisco, pp. 30-37, 1995.

[2] Bazzan A., "A Distributed Approach for Coordination Between Traffic Signals," *in Proceedings of 12th Annual Conference of Computer Society of Iran*, Iran, pp. 131-164, 2005.

[3] Bazzan A., "Traffic Signal Coordination Based on Distributed Problem Solving," *in Proceedings of IFACnFORS Symposium on Transportation Systems Theory and Application of Advanced Technology*, China, pp. 957-962, 1994.

[4] Bitting E. and Ghorbani A., "Cooperative Multiagent Systems for the Optimization of Urban Traffic," *in Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, China, pp. 176-182, 2004.

[5] Chee C., Srinivasan M., and Cheu L., "Cooperative Hybrid Agent Architecture for Real-Time Traffic Signal Control Systems," *Journal of Part a Systems and Humans, IEEE Transaction Man and Cybernetics*, vol. 33, no. 5, pp. 597-607, 2003.

[6] Chen W. and Decker K., "Developing Alternative Mechanisms for Multiagent Coordination," *in Proceedings of Lecture Notes in Computer Science*, Heidelberg, pp. 59-78, 2002.

[7] Haykin S., *Neural Networks, a Comprehensive Foundation*, Prentice Hall, 1999.

[8] Pan G. and Maddox A., "A Framework for Distributed Reinforcement Learning," *in Proceedings of Lecture Notes in Computer Science*, Berlin, pp. 97-112, 1995.

[9] Ricci A., Omicini A., and Denti E., "Objective vs. Subjective Coordination in Agent-Based Systems a Case Study," *in Proceedings of Lecture Notes in Computer Science*, Heidelberg, pp. 291-299, 2002.

[10] Roozemond D., "Autonomous Urban Traffic Control Intelligent Transport Systems," *in Proceedings of 4th World Congress an Intelligent Transport Systems*, Germany, pp. 1-5, 1997.

[11] Sanchez M. and Lucas W., "Exploring the World of Agent-Based Simulations: Simple Models, Complex Analyses," *in Proceedings of the Simulation Conference*, Florida, pp. 116-126, 2002.

[12] Sascha O., Cuena J., and Garcia A., "A Case of Multiagent Decision Support: Using Autonomous Agents for Urban Traffic Control," *in Proceedings of Lecture Notes in Artificial Intelligence*, Heidelberg, pp. 468-469, 1998.

[13] Srinivasan D. and Choy C., "Cooperative Multi-Agent System for Coordinated Traffic Signal Control," *IEEE International Transportation System*, vol. 153, no. 1, pp. 41-50, 2006.

[14] Sutton R. and Barto A., *Reinforcement Learning an Introduction*, Press Cambridge, 1998.

[15] Tan H. and Kwok H., "Applying Intelligent Agent Technology as the Platform for Simulation," *in Proceedings of the Simulation Symposium*, Boston, pp. 180-187, 1998.

[16] Xiac-Ming L. and Fei-Yue W., "Study of City Area Traffic Coordination Control on the Basis of Agent," *in Proceedings of the IEEE Intelligent Transportation Systems Conference*, Singapore, pp. 758-761, 2002.

[17] Zhongzhi Sh., "The Advanced Artificial Intelligence," *in Proceedings of Beijing Science Press*, China, pp. 223-226, 1998.

Shahaboddin Shamshirband currently, is a PhD student at university of Malaya. He worked as a senior lecturer in artificial intelligence at the University of Chalous, Iran. He holds Msc degree in computer science from Mashhad University, Iran in 2007.