Multi-intersections Traffic Signal Intelligent Control Using Collaborative Q-learning algorithm

Li Chun-gui
Department of Computer Engineering
Guangxi University of Technology
Liuzhou, China

Yan Xiang-lei Lin Fei-Ying Zhang Hong-lei
Dept. of Electronic Information and Control Engineering
Guangxi University of Technology
Liuzhou, China

Abstract—**Since congestion of traffic is ubiquitous in the modern city, optimizing the behavior of traffic lights for efficient traffic flow is a critically important goal. However，agents often select only locally optimal actions without coordinating their neighbor intersections. In this paper, an urban road traffic area-wide coordination control algorithm based on collaborative Q-learning is proposed. The agent model of traffic intersections is demonstrated. The algorithm substantially reduces average vehicular delay by using a collaborative Q-learning algorithm and can cooperative control of multiple intersections to achieve a near optimal control policy. The computer simulation results show that the control algorithm can effectively reduce the average delay time and play a very good control effect with multi-intersections, so the coordination method used in this paper is effective.**

Keywords- Q-learning; Traffic signal; Coordination control; Intelligent Transportation Systems

I. INTRODUCTION

With the increase of the number of vehicles and the density of traffic networks，mutual influence of traffic flows among adjacent intersections is gradually strong. Correlation among intersections has become increasingly evident. How to design traffic signal coordinated control efficiently from a system point of view, has become a new requirement for the development of traffic control. As urban traffic signal system has multiple information sources, time variability and randomness characteristics, so it is difficult to control them effectively with traditional theories and methods [1]. Therefore some advanced control theories and intelligent methods applied to urban traffic control. Researchers have applied various artificial intelligence techniques such as neural network [2], Fuzzy logic [3] to develop better traffic signal control systems. But these methods have complex structure which makes it difficult to achieve better results, thus the control results of such traffic methods are also limited.

Over the years, for the cooperative traffic signal control, the green wave band methods have been studied [4]. But these method asks that the traffic flows keep uniform speed so that a better control solution can be achieved, so it is difficult to be used in the traffic networks and urban traffic which has a lot of intersections connected each other, these studies above have mostly been committed to an isolated intersection, which makes it difficult to improve the traffic capacity of the entire road network effectively.

Since traffic control is fundamentally a problem of sequential decision making, and at the same time is a task that is too complex for straightforward computation of optimal solutions or effective hand-coded solutions, it is perhaps best suited to the framework ofMarkov Decision Processes (MDPs) and reinforcement learning (RL) or approximate dynamic programming (ADP), in which an agent learns from trial and error via interaction with its environment[5].

Reference [6] use approximate dynamic programming to replace the traditional methods, through this we can essentially overcome the deficiencies of traditional dynamic programming and alleviate the computer storage burden. However, the method is only for isolated intersection, without the cooperation among adjacent intersections, the near optimal control for each individual intersection can not guarantee a larger traffic area composing several intersections to be near optimal; Reference [7] propose artificial neural network for intelligent control of multiple intersections which can be effectively solve the urban traffic congestion problem, but because it use the gradient descent method training network parameters, so it is difficult to meet the random and uncertain urban transport flow. Reference [8] apply the dynamic programming for multi-intelligent intersection control, this method can be adjusted in real time according to the green light signals time, i.e. to overcome the drawbacks of fixed time. However, it is often unable to run true dynamic programming, because the DP recursively calculates Bellman's equation backwards step-by-step to find the optimal action that transfers the system from the current state to a new state, it is demand more the state space and the information space, i.e., as a result of the well-known "curse of dimensionality". Furthermore, the DP requires complete information on the time period in which the controller seeks optimization in real-time operation. However, traffic detectors may supply only 5-10 s data of future arriving vehicles.

Based on the previous work, we proposed a collaborative Q-learning algorithm to control multiple intersections in this paper. Simulation results show that this method can reducing the average waiting time of vehicles effectively.

978-1-4244-9953-3/11/$26.00 ©2011 IEEE

II. TRAFFIC INTERSECTION MODEL

Intersection is an entity with autonomous decision-making, so a well-designed intersection can effectively achieve the merger and separation of the vehicle volume, also it's a key to make transport system running normally. Therefore traffic intersection model is a very important kind of Agent in urban traffic simulation system. The model's structure is shown in Figure.1.

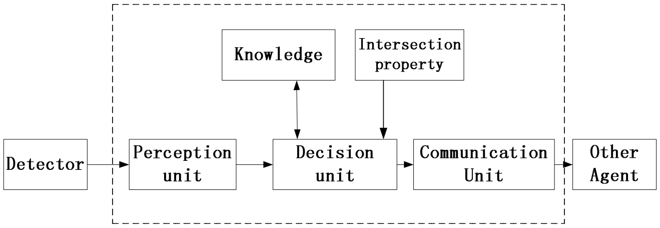


Figure 1. The agent-intersection model structure

Usually， reinforcement learning (RL) problems are modeled as Markov decision process (MDPs). These are described by a set of state *s,* a set of action *a*, a reward function $R(s,a)\to R$ and a probabilistic state transition function $P(s,a,s')\to[0,1]$. An experience tuple $\langle s,a,s',r\rangle$ denotes the fact that the agent was in state *s*, performed action *a* and ended up s' with reward *r*. $P(s,a,s')$ denotes the probabilistic that the agent uses action *a* to make environmental condition S migrated to the environmental condition s'. Agent's goal is find the optimal strategy in every discrete environmental state space in order to make the biggest discount reward and expectations [9].

Q-learning is a model-free approach to reinforcement learning which does not require the agent to have access to information about how the environment works. Q-learning works by estimating state-action values, the Q-value, which is a numerical estimator of quality for a given pair of state and action. More precisely, a Q-value Q(s,a) represents the maximum discounted sum of future rewards an agent can expect to receive if it starts in state s， choose action a and then continues to follow an optimal policy. Q-Learning algorithm approximates Q(s, a) as the agent acts in a given environment. The update rule for each experience tuple $\langle s,a,s',r\rangle$ is given by Eq.（1）where $\gamma\in(0,1)$ is the discount for future rewards.

Define Q value as estimate of state -action，According to the definition of MDPs：

$$Q(s,a)=R(s,a,s')+\gamma\sum_{s'\in S}P(s,a,s')\max_{a'}Q(s',a') \qquad （1）$$

Though Eq.（1）we can get the final Q value if known R and P condition because Q value is a sum of future reward. when the Q-values have nearly converged to their optimal values，the action with the highest Q-value for the current state can be selected.

III. A COLLABORATIVE Q-LEARNING ALGORITHEM

In MDPs, the transition of environment state is defined by transition probability function，which is not changing with time. In multiple Agent online learning environment, each Agent behavior is changed with its study circumstance. However, many established model based on MDPs is not doing more improvement on learning methods. In this paper, we propose to use an collaborative Q-learning algorithm based on current collaborative environment belief to replace the tradition Q-learning algorithm which based on single state action; the multiple Agent structure is shown in Fig.2.

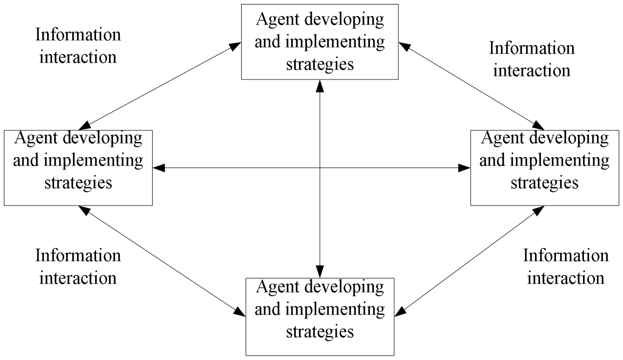

Information interaction
Agent developing and implementing strategies
Information interaction
Agent developing and implementing strategies
Agent developing and implementing strategies
Information interaction
Agent developing and implementing strategies
Information interaction

Figure 2. Distributed agent structure

The structure diagram consists of *n*-agent，An *n*-agent is a tuple (N,S,A,R,P) where：

- $N=1,.....i,...n$ is the set of agents.
- S is the discrete state space（set of *n*-agent stage games）.
- $A=\times A^i$ is the discrete action space （set of joint actions）.
- R^i is the reward function （R determines the payoff for agents i as $r^i:S\times A^1\times\cdots\times A^k\to R$ ）.
- P is the transition probability map （set of probability distributions over the state space S ）

If all agents keep mappings of their joint states and actions，this implies that each agent needs to maintain tables whose sizes are exponential in the number of agents：$|S^1|\times|S^2|\times\cdots\times|S^n|\times|A^1|\times|A^2|\times\cdots\times|A^n|$.This is hard even in the case of single state game where $|S|=1$. For example，assuming that agents playing the repeated game have only two actions，the size of the table is $2^{|N|}$. So it is demand more state space, information space and action space. Therefore in this paper we proposed a collaborative Q-learning algorithm to control multiple intersections which is through estimate the faith of opponent and environment knowledge instead of Q value function, so it need not observe the opponents reward and its Q learning parameters.

Then we adopt combination strategy (a_{self}, a_{other}) to estimate the state – actions Q value，and a_{self} represent the Agent's action, a_{other} represent the opponent Agent's action. In the current environment condition S, agent estimates that opponent Agent action and chooses a course of action based on Eq.（2）

$$P[a_{self} | s] = \frac{e^{Q(s, a_{self})/\tau}}{\sum_{a_k \in A} e^{Q(s, a_k)/\tau}} \qquad (2)$$

Where $P[a_{self} | s]$ is the probability that select a_{self} action in state s ； $Q(s, a_{self})$ is the $P_s = P[s' | s, a]$ expectation Q value function in state s . the reward r_t is denote the fact that in state s ， performed action a and ended up in s' . Agent updates Q value based on the following equation.

$$\begin{aligned} Q_t(s, a_{self}, a^*_{other}) &= (1 - a_t) Q_{t-1}(s, a_{self}, a^*_{other}) \\ &+ a_t (r_t + \gamma \max_{a'_{self} \in A} Q_{t-1}(s', a'_{self}, a'_{other})) \end{aligned} \qquad (3)$$

Where $a'_{other} = \arg\max_{a'_{other}} P_{s'}$ ， a^*_{other} is the opponent agents action ， $a_t \in [0,1]$ which is the learning rate at time *t*. a_t declined with time so as to facilitate the learning algorithm convergence.

With the process of learning，the Q value converge to the optimal value Q^* and the agent's knowledge tends to be precise. Then if we still make lots of exploration activities, the system performance maybe decline，so when the agent take enough environment knowledge，it is necessary to reduce exploration. So we introduce recently explore surplus to Q-learning，the equation can be written as

$$\begin{aligned} Q'_t(s, a_{self}, a^*_{other}) &= Q_t(s, a_{self}, a^*_{other}) \\ &+ \sigma\lambda(p_t(s, a_{self}, a^*_{other}))\sqrt{p_t(s, a_{self}, a^*_{other})} \end{aligned} \qquad (4)$$

where p_t is the waiting time surplus ， $b(p_t) = \sigma\lambda(p_t(s, a_{self}, a^*_{other}))\sqrt{p_t(s, a_{self}, a^*_{other})}$ is the explore surplus，through Eq.（4）the learning agent just only explore those did not reached state，by doing this，we can accelerate the learning process and reduce the state space.

For all agent，through observe state，action and reward，we can compute the average reward $\bar{r}$ ，and the reward are updated by

$$r_t = a_t \times \bar{r} + (1 - a_t) \times \bar{r}_{old} \qquad (5)$$

The learning rate a_t are updated by

$$a_t = \frac{a_0}{n_t(s, a_{self}, a^*_{other})} \qquad (6)$$

Where $n_t(s, a_{self}, a^*_{other})$ means the agent gain the experience number until t moment.

The traffic signal cooperative control algorithm using the collaborative Q-learning algorithm can be summarized as the following (for each signal controller):

Step1: Initial system state $\forall s \in S$, $\forall a \in A$, $Q_{0(s,a)} = 1$ ， set a_0, r .

Step 2: In the current environment condition s ， form all possible action a ， agent select the optimal action a_{self} according equation（2）based on the believe P_s .

Step 3：Carry out the optimal action a_{self} ， get action a^*_{other} from opponent agent when environment change to new state s' in the point t .

Step 4：Agent calculate reward value r_t based on equation（5）and update learning rate a_t based on equation（6）.

Step 5: Update $Q_t(s, a_{self}, a^*_{other})$ based on equation（3）and equation（4）.Then storage Q value.

Step 6：If the Agent unsatisfied the current proposal，then accord the opponent information choose a new action a' ，go to step 2；Otherwise，stop.

IV. EXPERIMENTS AND RESULTS

The experiments presented in this paper were conducted using the platform based on The Green Light District (GLD) traffic simulator [9], which is wrote by java. The cooperative signal controller is adopted for every intersection. GLD is a microscopic traffic model, i.e., it simulates each vehicle individually, instead of simply modeling aggregate properties of traffic flow. For the experiments，a small network of six intersections is used, each road has four lanes, the ratio of left, straight, and right is 1:2:1, traffic flow is given by Poisson distribution with a mean λ. For comparison, uniformly random policy；longest queen policy；Collaborative Q-learning policy signal controller is applied. The minGreen time is 15*s*, maxGreem time is 50*s*, the green extension is 20*s*, the max detectable number of vehicles is 30, and total time can not exceed 200*s*.

We consider a case of two traffic intersections. The model is shown in Figure 3. The intersection 3 behalf of East Central road and West Wenchang road, intersection 4 behalf of East Central road and Tanzhong road. There are four phases (straight-going in east-west, left-turning in east-west, straight-going in south-north, left-turning in south-north) in each intersection. The right-turning is not considered in this paper.

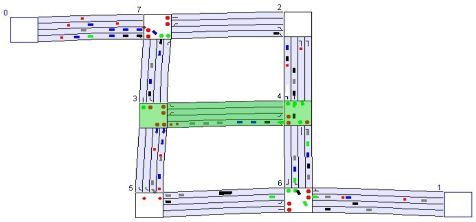
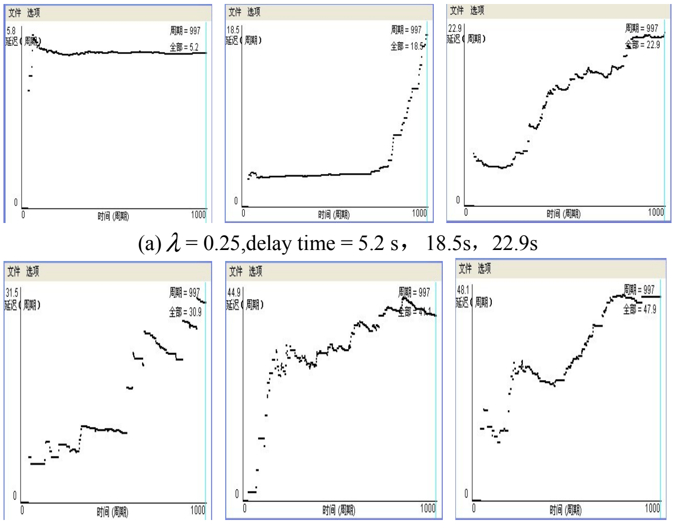


Figure 3. Road Network Model

Simulation program runs about 1000 traffic adjust cycles. The simulation results are the average delay time. In this experiment, we compared results with different λ; the results have been shown in Figure 4.



(a) $\lambda = 0.25$,delay time = 5.2 s， 18.5s，22.9s

(b) $\lambda = 0.8$, delay time = 30.9 s， 41.1s，47.9s

Figure 4. Average delay time in different traffic state

In figure.4, the left column plots show the average delay time with uniformly random policy, the middle column plots show the average delay time with longest queen policy and the right column plots show the average delay time with collaborative Q-learning policy.

Table 1 Average delay time of three signal control algorithm in different traffic flow

	$\lambda = 0.25$	$\lambda = 0.8$
collaborative Q-learning policy.	5.2	30.9
longest queen policy	18.5	41.1
uniformly random policy	22.9	47.9

From the results we can learn that when the traffic flow is low(λ=0.25), delay time can reduces 75% between uniformly random policy and corporation using collaborative Q-learning policy control, shown that the method proposed in this paper can effectively reduce the delay time; With the increase traffic flow (λ=0.8), this control effect is not obvious, the reason is in this method we only due to a two-simulation model, when the traffic flow is high even saturated, it is need to broaden road network, only this can we reduce the average delay time.

V. CONCLUSION

This paper investigates the application of a collaborative Q-learning policy for multi-intersections signal control in city. We show in the experiments that the Q-learning policy controllers meet all of the objectives; the key feature of the collaborative Q-learning policy approach is to estimate the faith of opponent and environment instead of Q value function. The simulation experiments show that the method can effectively reduce the average waiting time, therefore the Q-learning policy controller approach is a practical candidate for real-time at multi-intersections.

During the learning, the presented controller considers not only its own performance but those in the neighbor intersections. Next step is to promote controller algorithm to urban traffic networks, optimized the entire transport to make it become more and more smooth.

VI. ACKNOWLEDGMENT

This paper is supported by the Nature Science Foundation of Guangxi Zhuang Autonomous Region of China under Grant No.0481016 & The Scientific Research and Technology Development Programming Foundation of Guangxi Zhuang Autonomous Region of China under Grant No.0992006-13.

REFERENCES

[1] Z.Y.Liu, Intelligent Traffic Control Theory and Application. Beijing: Science Press, 2003, pp. 164-222.

[2] M. Saito and J. Fan, "Artificial neural network-based heuristic optimaltraffic signal timing," Comput.-Aided Civ. Infrastructure. Eng, 2000, pp. 281–291.

[3] M. B. Trabia, M. S. Kaseko, and M. Ande, "A two-stage fuzzy logic controller for traffic signals," Transp. Res. C, Emerg. Technol. 1999, pp. 353–367.

[4] G. J. Shen, "Urban traffic trunk two-direction green wave intelligent control strategy and its application," in proc.of the 6th World Congress on Intelligent Control and Automation, 2006, pp. 8563-8567.

[5] Bram Bakker, Shimon Whiteson, Leon Kester, and Frans Groen, "Traffic Light Control by Multiagent Reinforcement Learning Systems", Interactive Collaborative Information Systems, Studies in Computational Intelligence, pp. 475–510, Springer, Berlin, Germany, 2010.

[6] Chen Cai, Chi Kwong Wong, Benjamin G. Heydecker. "Adaptive traffic signal control using approximate dynamic programming," Transportation Research, Part C, 2009, no. 17 pp. 456-474.

[7] Tao Li, Dongbin Zhao, Jiangqiang Yi, "Adaptive Dynamic Programming for Multi-intersections Traffic Signal Intelligent Control," 11th International IEEE Conference on Intelligent Transportation Systems, Beijing, 2008, pages: 286-291.

[8] Y.L.Li, "Urban Traffic Flow Prediction and Assignment with Dynamic Programming", Journal of Transportation Systems Engineering and Information Technology, 2009, Vol.9, No. 3, pp. 135-139.

[9] Ana L.C.Bazzan, Denise de Oliveira, Bruno C.da Silva, Learning in groups of traffic signals , Engineering Applications of Artificial Intelligence, 2010, No. 23, pp.560-568

[9] M. Wiering, Multi-Agent Reinforcement Learning for Traffic Light Control, in Proc.17th International Conf. on Machine Learning, 2000, pp. 1151-1158.