'90 KACC 1990. 10. 26~27

Decentralized Learning Automata for Control of Unknown Markov Chains

Motoshi HARA and Kenichi ABE
Department of Information and Computer Sciences,
Toyohashi University of Technology, Toyohashi, 440, Japan

Abstract

In this paper, we propose a new type of decentralized learning automata for the control of finite state Markov chains with unknown transition probabilities and rewards. In our scheme a β-type learning automaton is associated with each state in which two or more actions(desisions) are available. In this decentralized learning automata system, each learning automaton operates, requiring only local information, to improve its performance under local environment. From simulation results, it is shown that the decentralized learning automata will converge to the optimal policy that produces the most highly total expected reward with discounting in all initiall states.

1 Introduction

Stochastic learning automaton, which opetates in random enviroments, have been extensively studied over past two decades[1]. We have presented a new family of learning automata termed β-type , which indicates the property of conditional optimality under some stationary random enviroments[2]. Recently the major arguments on both the α-type and the β-type learning automata are focused on the decision makers which interact in a decentralized manner to overcome the dimensional difficulty[1][3]. For example, the decentralized control by using the α-type learning automata in computer and communication network has been demonstrated both from the practical point of view of convergence speed and computational efficiency[1]. We also have proposed a construction of decentralized learning systems by using the β-type learning automata to some probabilistic optimization problems, e.g. the shortest path problem in stochastic networks and the learning control of unknown Markov chains[4][5].

The learning control of finite Markov chains with unknown dynamics is widely applicable. The Markov decision process arises when state transitions generate rewards which depend upon decisions taken in some or all states. In case that prior dynamics of the chain are previously known, some methods, e.g. Howard's policy iteration method and so on, are effective to find the optimal policy which produces the most highly total reward in all initial states. However, in many real applications, we often have to consider the Markov chains with unknown dynamics, hence the learning control problem arises.

In general, this type of control has dimensional difficulty. That is, the computation becomes burdensome when the number of states is very large. Further, an adaptive control problem results from the ignorance of both transition probabilities and corresponding rewards associated with various actions. The adaptive control appoach, however, needs the estimation of those many parameters. One effective approach to this problem is to use learning automata as decentralized decision makers[7].

In this paper, we show how to construct the decentralized system of the β-type learning automata and apply it to the learning control of finite Markov chains with unknown transition probabilities and rewards. In our scheme, each component, a β-type learning automaton, of the decentralized system requires only the local information without a coordinator. At this point of view, this system is different from the known decentralized system by using α-type learning automata. Finally, some simulation results are presented.

2 Single β-type learning automaton in a Q-model stationary random environment[2]

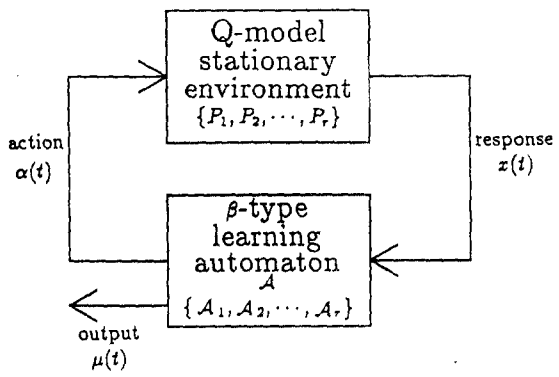

Fig.1 Automaton-Environment Configuration

In this study, we use the learning automaton termed β-type with the reinforcement scheme which is similar to the Baysian learning scheme. In this context, we term the learning automaton which is surveyed by K.S.Narendra et al.[1] "α-type". A learning automaton is a feedback system connecting an automaton , which chooses an action at each time, and an environment, which produces

responses to those actions (See Fig.1). We describe briefly the single β-type learning automaton model as follows.

1) Environment

In general, a random environment model has an action $\alpha(t) \in \alpha = \{\alpha_1, \alpha_2, \cdots, \alpha_r\}$ as its input ,where α is the action set of the learning automaton(r:action number), and a random variable $x(t) \in X = \{x_1, x_2, \cdots, x_n\}$ as its response , where X is the response set (n:response number). Here, we assume that X is a set of distinct real numbers which indicate the degrees of success i.e. , the rewards and $x(t)$ obeys unknown probability distribution P_i on X corresponding to an action $\alpha_i \in \alpha$. The random environment in which learning automaton operates is specified by a collection of r unknown probability distributions $P_i = (p_{i1}, p_{i2}, ..., p_{in})$ (i=1,2,...,r), which satisfy the following conditions;

$$0 \leq p_{ij} = p_r[x(t) = x_j | a(t) = a_i] \leq 1,$$

and

$$\sum_{j=1}^{n} p_{ij} = 1, \text{ for all i,j.}$$

In the case that 2<n< ∞ and the probability distributions P_i(i=1,2,...,r) are stationary, the environment is called as Q-model stationary environment.

2) A β-type learning automaton

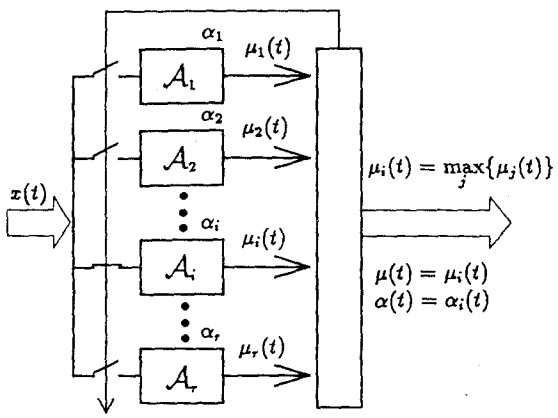

Fig.2 Construction of β-Type Learning Automaton

A β-type learning automaton $\mathcal{A}$ consists of r internal automata (See Fig.2) and each internal automaton $\mathcal{A}_i$ corresponds to an action $\alpha_i \in \alpha$. An internal automaton $\mathcal{A}_i$ is described by a 4-tuple <X, Ω,$\lambda_i(t)$, T>, where $\Omega = \{\omega_1, \omega_2, ..., \omega_m\}$ is the set of its states (m:state number), $\lambda_i(t) = (\lambda_{i1}(t), \lambda_{i2}(t), ..., \lambda_{im}(t))$ is the state probability vector at time t and T is the reinforcement scheme defined as below. The state probability vector satisfies the following conditions,

$$0 \leq \lambda_{ij}(t) \leq 1,$$

and

$$\sum_{j=1}^{m} \lambda_{ij}(t) = 1, \text{ for all i,j.}$$

For each automaton $\mathcal{A}_i$, we assign a real number μ_k to each state ω_k. These numbers μ_k(k=1,2,...,m) are taken to satisfy the following conditions:

$$\underline{x} < \mu_1 < \mu_2 < \cdots < \mu_m < \bar{x},$$

$$\underline{x} = \min_i(x_i) \quad and \quad \bar{x} = \max_i(x_i).$$

The interaction between a single β-type learning automaton and a Q-model stationary environment is stated as follows.

At the start time t=0, state probability vectors $\lambda_i(0)$(i=1,2,...,r) are set as

$$\lambda_{i1}(0) = \lambda_{i2}(0) = \cdots = \lambda_{im}(0) = \frac{1}{m}, \text{ for all i.} \quad (1)$$

At time t, each internal automaton $\mathcal{A}_i$ chooses its state $\omega_{ki} \in \Omega$ randomly according to its state probability vector $\lambda_i(t)$ and generates the output $\mu_i(t) = \mu_{ki}$. The β-type learning automaton chooses an action $\alpha(t) = \alpha_i \in \alpha$ corresponding to the internal automaton $\mathcal{A}_i$ which generates the most highly output and determines $\mu(t) = \mu_i(t)$ as the output. Thus, the β-type learning automaton performs action $\alpha(t)$ to the environment and subsequently receives the reward $x(t)$ as the environment response corresponding to $\alpha(t)$. So the state probability vector $\lambda_i(t)$ of the internal automaton $\mathcal{A}_i$ corresponding to the chosen action $\alpha(t) = \alpha_i$ is updated by

$$\lambda_i(t+1) = T(\lambda_i(t), x(t)). \quad (2)$$

In such scheme, the β-type learning automaton changes its probabilistic structure in order to adjust itself to the environment in iterative manner.

3) Reinforcement scheme

The reinforcement scheme T is defined as follows;

$$\lambda_{ik}(t+1) = c\lambda_{ik}^{x'(t)}(t)\{\mu_k(1-\mu_k)\}^{1-x'(t)}\}^{\theta}, \quad k=1,2,...,n \quad (3)$$

where c is the normalizing constant, θ is the parameter which dominates the convergence speed and $x'(t)$ is the normalized environment response, which lies in the interval [0,1], defined by

$$x'(t) = \frac{x(t) - \underline{x}}{\bar{x} - \underline{x}}. \quad (4)$$

Particularly when θ is equal to 1, the scheme is the same form as the Baysian learning scheme.

Here, let c_i be the expected reward under an action $\alpha_i \in \alpha$, which is defined as

$$c_i = E[x(t)|\alpha(t) = \alpha_i] = \sum_{j=1}^{n} p_{ij} x_j. \quad (5)$$

The value c_i is the evaluation of the action α_i and we assume that a unique maximum of c_i(i=1,2,...,r) exists. The ultimate goal of the learning automaton is to find the action that produces the most highly expected reward in iterative manner under the unknown random environment.

In evaluating the performance of the learning automaton, the concepts of optimality and ϵ-optimality are very important. Optimality means that a learning automaton chooses the optimal action or actions from its action set with probability one as time goes infinity and ϵ-optimality is a weaker concept than the optimality.

By using reinforcement scheme(3), the β-type learning automaton is optimal under certain condition on the random environment, and the output $\mu_i(t)$ of each internal automaton $\mathcal{A}_i$ converges asymptotically to the value which is the nearest to the expected reward c_i. Hence the output $\mu(t)$ of the learning automaton approaches asymptotically to the value which is the nearest to the expected reward under the optimal action.

Generally, the action number of the learning automaton is very large and the single learning automaton model is not appropriate neither from the practical point nor view of the convergence speed and computational efficiency. This is why that the major arguments on the learning automata are focused on the decision makers which interact in a decentralized fashion. Decentralized learning automata consists of a large number of learning auotmata which are interconnected and each learning automaton has only a few actions.

3 Learning control of unknown Markov chains

The control of Markov chains can be stated as follows.

Let $S = \{s_1, s_2, ..., s_N\}$ $(N < \infty)$ be the state space of a finite Markov chain and $D^i = \{d_1^i, d_2^i, ..., d_{r_i}^i\}(r_i < \infty)$ be the finite set of decisions available in state $s_i \in S$. At each time t=1,2,..., the system of the Markov chain consists in one state of the state space S, and whenever the system consists in state s_i one decision $d_{k_i}^i$ is always chosen from D^i. Suppose that the decision $d_{k_i}^i \in D^i$ is chosen in the state s_i at time t. Then the system goes from s_i to s_j according to a transition probability $q_{ij}^{k_i}$. Associated with the state transition from s_i to s_j, a reward $r_{ij}^{k_i}$ is generated , where

$$0 \le q_{ij}^{k_i} \le 1, \sum_j q_{ij}^{k_i} = 1$$

and

$$|r_{ij}^{k_i}| < \infty, \text{ for all i, j, } k_i$$

The goal of the control of finite Markov chains is to choose a policy which maximises the criterion of the total expected reward that will be generated from the present time. In the criterion, the future reward is assumed to be discounted per unit time by a discount factor β. Here, the policy consisting of one action at every state is denoted by N-dimensional vector $\equiv (d_1^i, d_2^i, \cdots, d_{k_N}^i) \in \mathcal{D}$, where $\mathcal{D} = D^1 \otimes D^2 \otimes \cdots \otimes D^N$ is the set of plicies.

If the system is in state s_i at the present time, the criterion under the policy P becomes

$$V_\beta(i, P) = r_i^{k_i} + \beta \sum_j q_{ij}^{k_i} V_\beta(j, P). \quad (6)$$

The quantity $r_i^{k_i}$ is the expected reward from a single transition from state s_i under decision $d_{k_i}^i$, thus $r_i^{k_i}$ is obtained as

$$r_i^{k_i} = \sum_j q_{ij}^{k_i} r_{ij}^{k_i}. \quad (7)$$

If $P^* = (d_{k_1}^{1\,*}, d_{k_2}^{2\,*}, \cdots, d_{k_N}^{N\,*})$ is the optimal policy that maximises $V_\beta(i, P)$ for each i, P^* satisfies the following condition;

$$V_\beta(i, P^*) \ge V_\beta(i, P), \text{for all i, every } P^* \ne P. \quad (8)$$

In order to find the optimal policy P^*, some effective methods, e.g. the Howard's policy iteration method[6] and so on, have been presented if the prior dynamics of the chain are previously known. However, in the practical cases, we often have to consider the Markov chains with unknown dynamics, e.g. transition probabilities and rewards. In this case, since the unknown information about the system must be learned for control decision, the learning control problems arise. And this type of control faces several difficulties with ragard to practical application.

First, in centralized approaches, the computational cost of the scheme increases dramatically with increasing N even if full information was previously given. In this respect, decentralization is highly desirable.

Second, in the case that the prior dynamics, transition probabilities and rewards, are unknown, the optimal policy cannnot be found off-line even if the computational problem can be overcome. Hence, the adaptive problems that needs the estimation of unknown parameters arise. One effective approach to the learning control of unknown Marcov chains is to use learning automata as decentralized decision makers[7].

4 Decentralized learning automata approach to the control of unknown Markov chains

As mentioned in section 3, the control scheme of finite Markov chains with unknown transition probabilities and rewards is implemented in a decentralized fashion. Recently, we have proposed a construction method of decentralized learning systems by using β-type learning automata to the shortest path problem in stochastic networks, and showen simultaneously the efficiency of the

method through some simulation results. Further, the method can be easily applied to the learning control of unkown Markov chains, which is described by the following algorithm(See Fig.3).

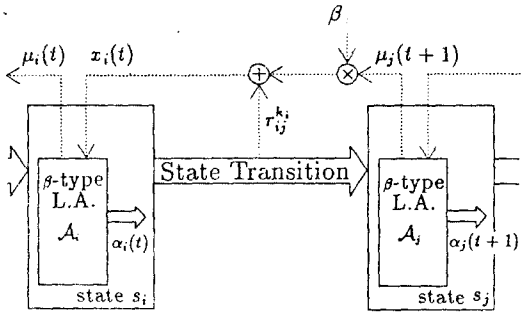

Fig.3 Construction of Decentralized Learning Automata

Algorithm

Step1. Start:

1) A β-type learning automaton as decentralized decision maker is associated with each state in which two or more decisions are available and operates only when the system of the Markov chain consists in that state. At state s_i, the learning automaton is denoted by $\mathcal{A}_i$, in which the action set and the output are defind as $\alpha^i = \{\alpha_1^i, \alpha_2^i, ..., \alpha_{r_i}^i\}$ and $\mu_i(t)$ respectively. In $\mathcal{A}_i$, each internal automaton of $\mathcal{A}_i$ is denoted by $\mathcal{A}_{ik}$(k=1,2,...,r_i), in which output at time t is denoted by $\mu_{ik}(t)$. And each element α_ν^i of α^i corresponds to the decision $d_\nu^i \in D^i$.

2) The initial state(at the time t=0) of the Markov chain is set arbitrary.

Step2. Iterate:

3) It is assumed that the Markov chain consists in the state s_i at the present time t. The automaton $\mathcal{A}_i$ chooses its action $\alpha_i(t) = \alpha_{k_i}^i \in \alpha^i$ and generates the output $\mu_i(t)$, that is this operation is equivalent to choosing the decision $d_{k_i}^i$ from D^i. This results in a transition from the state s_i to a new state s_j according to the transition probability $q_{ij}^{k_i}$ and a generation of the reward $r_{ij}^{k_i}$.

4) It is assumed that ,at the next time t+1, the state consists in s_j and $\mathcal{A}_j$ chooses an action(decision) from its action set(decision set) and generates the output $\mu_j(t+1)$. A transition from the state s_j occurs.

5) Then, the response $x_i(t)$ to the learning automaton $\mathcal{A}_i$ is given by

$$x_i(t) = r_{ij}^{k_i} + \beta \cdot \mu_j(t+1). \tag{9}$$

6) $t \leftarrow t+1$

In above algorithm, the iteration process consists of 3), 4), 5)and 6), but time t elapses in serial order with every iteration. The proposed decentralized learning automata select the optimal policy under this algorithm as time goes infinity. Thus, the decentralized learning automata can be represented by the simple network of several learning automata. Each component of the decentralized automata, as the decentralized decision maker, operates without the coordinator and needs only the local information to improve its performance under local environment. From this point of view, this decentralized learning automata is desirable as the autonomous decentralized system comparing with the decentralized system by using α-type learning automata.

Generally, there are two classifications in decentralized learning automata[1].

(a)Synchronous Models

In this class of models, each interconnected learning auotmaton of decentralized automata chooses an action at the same time and subsequently receives a environment response.

(b)Sequential Models

In contrary, only one learning automaton chooses an action at a time t. Then the chosen action determines who acts at next time and the response of local environment is generated.

From above classifications, our scheme is classified into "Sequential Models" according to the structure of its interconnection.

5 Simulation results

In this chapter, we introduce the evaluation function of the behavior of the decentralized learning automata described in section 4.

First, the evaluation function $f_T(t)$ is defined as

$$f_T(t) \equiv \frac{1}{T} \sum_i N_{iT}(t), \quad 1 < T < \infty,\ t \geq T \tag{10}$$

where $N_{iT}(t)$ indicates how often the decision $d_{k_i}^{i\,*} \in P^*$ is chosen by the learning automaton $\mathcal{A}_i$ at state s_i in the period from time t-T to time t. Obviously $f_T(t)$ lies in the interval [0,1] and indicates the percentage of the numbers of the optimal choises in the period. If $f_T(t)$ converges asymptotically to 1 , the decentralized learning automata will become to select the optimal policy from $\mathcal{D}$, therefore $f_T(t)$ can evaluate the collective behavior of decentralized learning automata on examination about the optimality.

Second, we introduce the evaluation function $e_T(t)$ of the estimation property of the learning automata. In our scheme, it is hard to analyse theoretically the asymptotic collective behavior of the estimation. However, if the decentralized learning automata select the optimal policy on basis of accurate estimation of the total expected

rewards with discounting, the output of each learning automaton $\mathcal{A}_i$ converges asymptotically to $V_\beta(i, P^*)$. Consequently, the output of each internal automaton will converge to certain value, which appears at last stage in the Howard's policy iteration method. $e_T(t)$ is defined by

$$e_T(t) \equiv \frac{\sum_{i,k_i} |\mu^T_{ik_i}(t) - V_\beta(i, k_i P^*)|}{\sum_{i,k_i} |V_\beta(i, k_i P^*)|}. \quad 1 < T < \infty, t \geq T \tag{11}$$

In (11), $\mu^T_{ik_i}$ denotes the average of the output $\mu_{ik_i}(t)$ of internal automaton $\mathcal{A}_{ik_i}$ in the period from time t-T to time t, $V_\beta(i, k_i P^*)$ is given by

$$V_\beta(i, k_i P^*) = r_i^{k_i} + \beta \sum_m q_{im}^{k_i} V_\beta(m, P^*), i = 1, 2, ..., N. \tag{12}$$

Note that $V_\beta(i, k_i P^*)$ coinsides with $V_\beta(i, P^*)$ only if $d^i_{k_i}$ is equal to $d^i_{k_i^*} \in P^*$.

In the Howard's policy iteration method, the optimal policy P^* is obtained in the last stage. So, in the last stage, the most highly total expected reward with discounting is given by

$$V_\beta(i, P^*) = \max_{k_i}\{V_\beta(i, k_i P^*)\}. \quad i=1,2,...,N, k_i=1,2,...,r_i \tag{13}$$

In the simulation works, $V_\beta(i, k_i P^*)$ is previously obtained off-line by the Howard's policy iteration method.

For computer simulation of implementing our scheme described in sectin 4, we provide two models specified in Table I and Table II. Table I describes the most simplest model of the Markov chain with two states, and Table II describes the Markov chain with five states.

In the former, the number of the policies denoted by $|\mathcal{D}|$ is $2^2 = 4$ and the parameter values used in producing results Fig.4 are β=0.3, m=100, $\underline{x}$=-100.0, $\bar{x}$=100.0 and θ=1.

In the later, the number of the policies is $3^5 = 243$, and the parameter values used in producing results Fig.5 are β=0.8, m=100, $\underline{x}$=-10.0, $\bar{x}$=10.0 and θ=1.

Table I

state	decision	transiton probabilities		rewards	
s_i	d^i_k	q^k_{ij}		r^k_{ij}	
	→ j	1	2	1	2
1	1	0.50	0.50	9	3
	2	0.80	0.20	4	4
2	1	0.40	0.60	3	-7
	2	0.70	0.30	1	-19

Table II

state	decision	transiton probabilities					rewards				
s_i	d^i_k	q^k_{ij}					r^k_{ij}				
	→ j	1	2	3	4	5	1	2	3	4	5
1	1	0.1	0.2	0.2	0.2	0.3	1	-2	-4	-1	3
	2	0.2	0.2	0.2	0.2	0.2	-2	-7	2	8	-1
	3	0.1	0.1	0.1	0.3	0.4	-3	6	-1	-2	4
2	1	0.1	0.1	0.2	0.3	0.3	7	-4	1	-2	-8
	2	0.1	0.1	0.6	0.1	0.1	-5	-2	1	-2	5
	3	0.1	0.5	0.2	0.1	0.1	5	-1	-3	-7	0
3	1	0.1	0.4	0.2	0.2	0.1	3	1	2	4	-4
	2	0.3	0.2	0.2	0.2	0.1	3	0	-1	-3	7
	3	0.3	0.2	0.1	0.1	0.3	-6	1	-2	4	0
4	1	0.2	0.5	0.1	0.1	0.1	5	-4	3	1	-6
	2	0.3	0.1	0.1	0.4	0.1	3	2	-1	-3	-5
	3	0.2	0.3	0.3	0.1	0.1	4	1	-6	6	2
5	1	0.2	0.2	0.2	0.2	0.2	2	6	2	-1	3
	2	0.2	0.3	0.2	0.1	0.2	-5	1	-3	4	-4
	3	0.1	0.4	0.2	0.1	0.2	-3	5	2	-1	-5

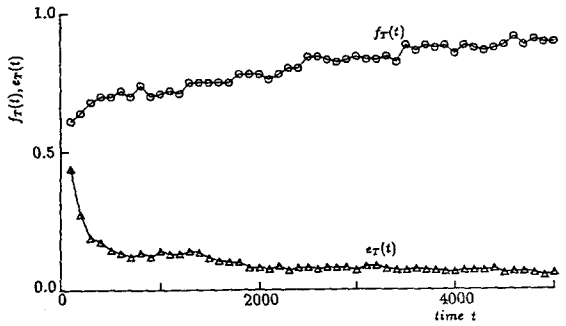

Fig.4 Variation curves both of $f_T(t)$ and $e_T(t)$(Table I)

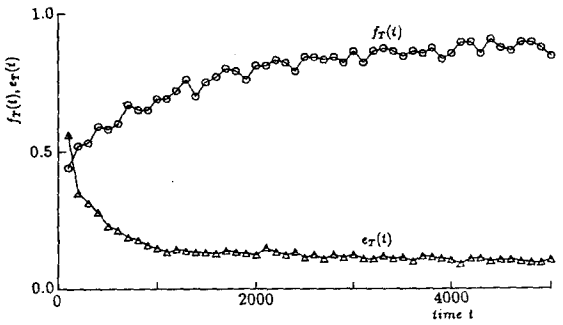

Fig.5 Variation curves both of $f_T(t)$ and $e_T(t)$(Table II)

Fig.4 and Fig.5 show that both $f_T(t)$ and $e_T(t)$ are averages of performance over the five runs. At the start of each run, we supplied a different seed value to the random number generator for both the state probabilities of learning automata and the transition probabilities of the Markov chain. Except for the random number generator seeds, identical parameter values are used for all runs. In

Fig.4 and Fig.5, it is shown that the proposed decentralized learning automata select the optimal policy in iterative manner, since $f_T(t)$ converges asymptotically to 1. And each $e_T(t)$ in those figures converges asymptotically to 0. Those results mean that each learning automaton controls the Markov chain on the basis of appropriate estimation of the total expected rewards which appear at the last stage in the Howard's policy iteration method.

6 Conclusion

An effective approach using the decentralized automata in decentralized fashion for the control of finite Markov chain with unknown transition probabilities and rewards has been proposed.

We have also examined the asymptotic properties of this scheme through two functions, $f_T(t)$ and $e_T(t)$. Our scheme has the following characteristics:

(i) The decentralized system shown in section 4 representes a simple network of several β-type learning automata.

(ii) The scheme is implemented without a coordinator. Each conponent operates depending upon only local information that is the output from one of the connected compornents in the network, and its control is localized. This fact motivates that our scheme is more desirable than the decentralized system using α-type learning automata with a coordinator. Because such a coordinator leads to the centralization of information.

(iii) Each internal automaton $\mathcal{A}_{ik_i}(k_i=1,2,...,r_i)$ of learning automaton $\mathcal{A}_i$ estimates appropriately the total expected reward $V_\beta(i, k_i P^*)$ that appears at the last stage in the Howard's policy iteration method.

The theoretic proof on the optimality of our scheme is still unkown, since the structure of the β-type learning automata that interact in a decentralized fashion for this problem is quite complicated. However, we have shown that, for the case of both Table I and Table II represented in section 5, satisfactory optimal control can be achieved using our scheme.

REFERENCES

[1] K.S.Narendra and R.M.Wheeler,Jr., in Adaptive and Learning System Theory and Applications(K.S.Narendra(ed)), Plenum Press,1986.

[2] K.Abe, On Conditional Optimality of A Class of Learning Automata in Random Environments , Information Sciences, vol.31, pp243-263, 1983.

[3] K.Abe, A Study of Decentralized Learning Automata (in Japanese), Proceedings of 20th ISCIE Symp. Stochastic Systems Theory and Its Applications, pp.75-79, 1988.

[4] M.Hara and K.Abe, Learning Automata Approach to Shortest Path Problems in Stochastic Networks (in Japanese), Proceedings of 21th ISCIE Symp. Stochastic Systems Theory and Its Applications, pp.67-70, 1989.

[5] M.Hara and K.Abe, Decentralized Learning Control of Finite Markov Chains (in Japanese), Proceedings of 28th SICE Annual Conference, pp.519-520,1990.

[6] R.A.Howard, Dynamic Programming and Markov Processes, Technology Press and Wiley, 1960.

[7] R.M.Wheeler,Jr. and K.S.Narendra, Decentralized Learning in Finite Markov Chains, IEEE trans. Automatic Control, vol.AC-31, No.6, 1986.