2-310 A Consideration on the Learning Behaviors of the Hierarchical Structure Learning Automata Operating in the Nonstationary S-model Random Environment

○ Norio Baba†, and Yoshio Mogami‡

† Dep. of Information Science, Osaka Kyoiku Univ., Kashiwara City, Osaka Prefecture, 582-8582, JAPAN
‡ Dep. of Information Sci. and Intelligent Sys., Fac. of Engineering, Univ. of Tokushima, Tokushima, 770-8506, JAPAN

Learning behaviors of hierarchical structure learning automata (HSLA) operating in the nonstationary S-model environment are considered. It is shown that an extended algorithm of relative reward strength algorithm ensure convergence to the optimal path with probability 1. Several computer simulation results confirm the effectiveness of the proposed algorithm. Further, learning behaviors of HSLA under the nonstationary multiteacher environment are also considered.

Key Words: Unknown Nonstationary Environment, Hierarchical Structure Learning Automata, Relative Reward Strength Algorithm

1. Introduction

After the pioneering work of Tsetlin[1], the study of learning automata (LAs) has been done quite extensively by many researchers [2]-[26]. LAs have made a significant impact upon many areas of engineering problems; and have so far been successfully applied to many areas. They are expected to provide one of the most powerful tools for constructing an intelligent system.

Although the study of LAs has matured, there are still several problems to be settled. Two of the most important are the low speed of learning, and the insufficient tracking ability to the changing environment (nonstationary environment). In order to overcome the first problem, the concept of the hierarchical structure automata was originally proposed by Thathachar and Ramakrishnan [8] and Ramakrishnan [9]. Since then, the learning behaviors of the hierarchical structure automata have been extensively studied by many researchers [5],[6],[10]-[12]. To overcome the second problem, Thathachar and Sastry [13] proposed a learning algorithm which uses the average of all the reward responses from the environment. They also extended the use of their algorithm to the hierarchical LAs model [14],[15]. Following their research, Oommen and Lanctot [16] and Papadimitriou [17] introduced the two concepts of discretized pursuit algorithm, and stochastic estimator learning algorithm. Their continuous work has yielded fruitful results [18],[19].

In 1989, Simha and Kurose [20] derived a very interesting algorithm whose approach is considerably different from previous algorithms. They proposed the relative reward strength algorithm, which utilizes the most recent reward response from the environment in an intelligent way. They proved that the proposed algorithm converges to the optimal action with probability 1. They also gave several computer simulation results which confirmed the effectiveness of their algorithm, and touched upon the possibility of using it efficiently in a certain type of nonstationary environment.

However, despite the effectiveness of their algorithm, additional studies concerning the learning behaviors of this type of algorithm have not followed. The present writers cannot explain such omission.

One of the most reasonable ways to use this type of algorithm in an environment with high dimensionality is to utilize the hierarchical system of the LAs. This means that one should extend the original relative reward strength algorithm [20] to be utilized in the hierarchical system of the LAs, and then carefully investigate its learning performance. However, unfortunately, this has not yet been attempted.

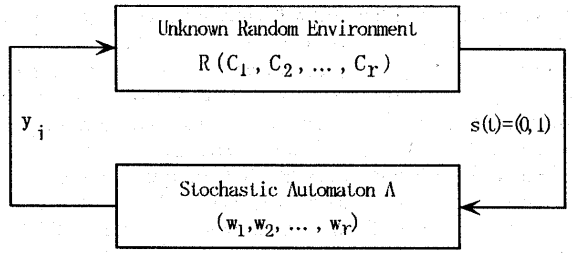

Fig. 1 Basic model of a learning automaton operating in an unknown random environment.

In this paper, we shall try to extend the algorithm of Simha and Kurose to be used in the hierarchical structure LAs model. We shall offer evidence that our extended algorithm converges to the optimal action under the certain type of the nonstationary S-model environment. We shall also give several computer simulation results which confirm the effectiveness of our extended algorithm. Further, we shall touch upon the learning behaviors of HSLA under the nonstationary multiteacher environment.

2. Basic Model of a Learning Automaton Operating in an Unknown Environment

Below we will discuss the learning behaviors of the HSLA operating in the nonstationary S-model environment. To place our study in the appropriate context, we start with a brief explanation of the basic model of the learning mechanism of the single automaton in the stationary random environment.

The learning behaviors of a variable-structure learning automaton operating in an unknown random environment have been discussed extensively under the basic model shown in Fig.1. Let us briefly explain the learning mechanism of the learning automaton A under the unknown random environment (teacher environment) $R(C_1, C_2, \ldots, C_r)$.

The learning automaton A is defined by the sextuple $\{S, W, Y, g, P(t), T\}$. S denotes the set of two inputs (0,1), where 1 indicates the reward response from $R(C_1, C_2, \ldots, C_r)$, and 0 indicates the penalty response. (If the set S consists of only the two elements 0 and 1, the environment is said to be a P-model. When the input into A assumes a finite number of values in the closed interval

[0,1], it is said to be a Q-model. An S-model is one in which the input into A takes an arbitrary number in the closed line segment [0,1]. In this paper, we will deal with the S-model environment.) W denotes the set of r internal states $(w_1, w_2, \ldots, w_r)$. Y denotes the set of r outputs $(y_1, y_2, \ldots, y_r)$. g denotes the output function $y(t) = g[w(t)]$, that is, one to one deterministic mapping. $P(t)$ denotes the probability vector $[p_1(t), p_2(t), \ldots, p_r(t)]'$ at time t, and its ith component $p_i(t)$ indicates the probability with which the ith state w_i is chosen at time t $(i = 1, 2, \ldots, r)$:

$$p_1(0) = p_2(0) = \cdots = p_r(0) = \frac{1}{r}, \quad \sum_{i=1}^{r} p_i(t) = 1.$$

T denotes the reinforcement scheme which generates $P(t+1)$ from $P(t)$.

Suppose that the state w_i is chosen at time t. Then, the learning automaton A performs action y_i on the random environment $R(C_1, C_2, \ldots, C_r)$. In response to the action y_i, the environment emits output $s(t) = 1$ (reward) with probability $1 - C_i$; and output $s(t) = 0$ (penalty) with probability C_i $(i = 1, 2, \ldots, r)$. If all of the C_i $(i = 1, 2, \ldots, r)$ are constant, the random environment $R(C_1, C_2, \ldots, C_r)$ is said to be a stationary random environment. (The term "single teacher environment" is also used.) On the other hand, if C_i $(i = 1, 2, \ldots, r)$ are not constant, it is said to be a nonstationary random environment. Depending upon the action of the learning automaton A and the environmental response to it, the reinforcement scheme T changes the probability vector $P(t)$ to $P(t+1)$.

The values of C_i $(i = 1, 2, \ldots, r)$ are not known in advance. Therefore, it is necessary to reduce the average penalty,

$$M(t) = \sum_{i=1}^{r} p_i(t) C_i$$

by selecting an appropriate reinforcement scheme. To judge the effectiveness of a learning automaton operating in a stationary random environment $R(C_1, C_2, \ldots, C_r)$, various performance measures such as optimality, ε-optimality, absolute expediency, etc. have been set up. (Details are omitted due to space; see [3]–[7].)

In this section, we have briefly introduced the learning mechanism of the single automaton under the stationary random environment $R(C_1, C_2, \ldots, C_r)$. However, when applying LAs to various actual problems, one often encounters hard situations where nonstationary random environment, multi-teacher environment, HSLA model, etc. must be considered. In the following section, we will explain the learning mechanism of the HSLA operating in the nonstationary S-model environment.

3. Hierarchical Structure Learning Automata(HSLA) Operating in the Nonstationary S-model Environment

Learning behaviors of the single automaton have been extensively studied under the basic model shown in Fig.1. However, one of the most serious bottlenecks concerning the learning model of the single automaton is that its learning performance declines considerably when the space of decisions has high dimensionality.

In order to overcome this problem, Thathachar and Ramakrishnan [8] proposed the concept of the HSLA. Since then, many active researchers have been involved in the study of the learning behaviors of the HSLA.

Next we consider the learning behaviors of the HSLA operating in the nonstationary S-model environment.

Hierarchical Structure Learning Automata

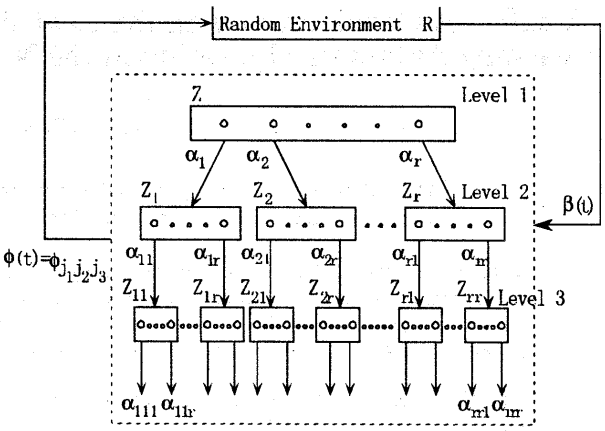

Fig. 2 Hierarchical structure learning automata.

The system of the hierarchical structure learning automata (HSLA) considered in this paper is briefly described as follows. (For easier understanding, the system of the HSLA with the hierarchy of 3 levels is shown in Fig.2.) The hierarchy consists of the tree-structured LAs each with r actions. Z denotes the top level (first level) automaton. $Z_1, Z_2, \ldots, Z_r$ denote the automata at the second level. $Z_{11}, Z_{12}, \ldots, Z_{rr-1}, Z_{rr}$ denote the automata at the third level. At the general sth $(s = 1, 2, \ldots, N)$ level, there are r^{s-1} automata $Z_{i_1 i_2 \cdots i_{s-1}}$.

Each learning automaton $Z_{i_1 i_2 \cdots i_{s-1}}$ $(s = 1, 2, \ldots, N)$ is defined by $\{\alpha_{i_1 i_2 \cdots i_{s-1}}, \beta(t), \boldsymbol{p}_{i_1 i_2 \cdots i_{s-1}}(t), T, \boldsymbol{v}_{i_1 i_2 \cdots i_{s-1}}(t)\}$. Here, $\alpha_{i_1 i_2 \cdots i_{s-1}} = \{\alpha_{i_1 i_2 \cdots i_{s-1} 1}, \alpha_{i_1 i_2 \cdots i_{s-1} 2}, \ldots, \alpha_{i_1 i_2 \cdots i_{s-1} r}\}$ denotes the set of r actions, $\beta(t)$ denotes the reward response from the environment, $\boldsymbol{p}_{i_1 i_2 \cdots i_{s-1}}(t) = (p_{i_1 i_2 \cdots i_{s-1} 1}(t), p_{i_1 i_2 \cdots i_{s-1} 2}(t), \ldots, p_{i_1 i_2 \cdots i_{s-1} r}(t))'$ denotes the action probability distribution, T denotes the reinforcement shceme, $\boldsymbol{v}_{i_1 i_2 \cdots i_{s-1}}(t) = (v_{i_1 i_2 \cdots i_{s-1} 1}(t), v_{i_1 i_2 \cdots i_{s-1} 2}(t), \ldots, v_{i_1 i_2 \cdots i_{s-1} r}(t))'$ denotes the recent reward vector [1].

Let us now examine how the hierarchical system behaves. Initially, all the action probabilities for each automaton at each level are set equal. The first level automaton Z chooses an action at time t from the action probability distribution $\boldsymbol{p}(t) = (p_1(t), p_2(t), \ldots, p_r(t))'$. Suppose that α_{j_1} is the output from Z. Then, at the second level, the automaton Z_{j_1} is actuated. The Z_{j_1} also chooses an action from its action probability distribution. Assume that $\alpha_{j_1 j_2}$ is the output from Z_{j_1}. Then, at the third level, the automaton $Z_{j_1 j_2}$ is actuated. This cycle of operation is repeated from the top (1st level) to the bottom (Nth level). The action at the lowest level interacts with the environment.

The sequence of actions $\{\alpha_{j_1}, \alpha_{j_1 j_2}, \ldots, \alpha_{j_1 j_2 \cdots j_N}\}$ having been chosen by N automata is called the path. Let $\phi_{j_1 j_2 \cdots j_N}$ denote the path. Corresponding to the path, the hierarchical structure learning automata model receives reward strength $\beta(t)$ [2] as an environmental response. The HSLA model utilizes this $\beta(t)$ in order to update the current recent reward vector. The action probability vector of each learning automaton relating to the path is updated by using the information concerning the recent reward vector. After all

[1] The recent reward vector will be defined in the next section.

[2] $\beta(t)$ can be an arbitrary number in the closed line segment [0,1]. The large value of $\beta(t)$ indicates that the high reward is given to the sequence of actions being chosen.

of the above procedures have been completed, time t is set to be $t+1$. By repeating the above cycle of operation, learning by the hierarchical structure automata proceeds. Let $\pi_{j_1 j_2 \cdots j_N}(t)$ denote the probability that the path $\phi_{j_1 j_2 \cdots j_N}$ is chosen at time t. From this definition,

$$\pi_{j_1 j_2 \cdots j_N}(t) = p_{j_1}(t) p_{j_1 j_2}(t) \cdots p_{j_1 j_2 \cdots j_N}(t) \tag{1}$$

The Nonstationary S-model Environment

Next, we consider the learning behaviors of the hierarchical structure automata model under the S-model nonstationary environment with the following property:

The optimal path $\phi_{j_1^* j_2^* \cdots j_N^*}$ satisfying the following relation exists uniquely [3] :

$$\min_t E\{\beta_{j_1^* j_2^* \cdots j_N^*}(t)\} > \max_t E\{\beta_{i_1 i_2 \cdots i_N}(t)\} \tag{2}$$

where $\beta_{i_1 i_2 \cdots i_N}$ denotes the environmental response corresponding to the path $\phi_{i_1 i_2 \cdots i_N}$ and $\max\{|j_1^* - i_1|, |j_2^* - i_2|, \ldots, |j_N^* - i_N|\} > 0$.

The objective for the hierarchical structure of LAs is to find the optimal path $\phi_{j_1^* j_2^* \cdots j_N^*}$ with the probability as high as possible.

Remark 1 The inequality (2) indicates that the minimum value of the average reward from the environment corresponding to the optimal path is the larger than the maximum value of the average reward corresponding to the other path. This means that we will discuss the learning performance of the proposed algorithm under the nonstationary S-model environment whose constraint is rather strict.

4. A New Learning Algorithm of the Hierarchical Structure Automata Model

In 1989, Simha and Kurose[20] proposed a new class of update algorithms which realize a sophisticated use of recent environmental responses. They showed that their algorithms are suitable for tracking nonstationary behaviors of environments. In this section, we shall extend their algorithms in order to use them effectively in the HSLA system.

First, let us define "recent reward to the path" and "recent reward vector":

Definition 1 Let us assume that the path $\phi_{i_1 i_2 \cdots i_N}$ has been chosen at time t and, corresponding to $\phi_{i_1 i_2 \cdots i_N}$, reward $\beta_{i_1 i_2 \cdots i_N}$ has been given from the environment. Then, "recent reward to the path $\phi_{i_1 i_2 \cdots i_N}$" is defined as follows:

$$u_{i_1 i_2 \cdots i_N}(t) = \beta_{i_1 i_2 \cdots i_N} \tag{3}$$

On the other hand, the other recent reward to the path $\phi_{j_1 j_2 \cdots j_N}$ ($^\exists k;\ i_k \neq j_k$) is defined as follows:

$$u_{j_1 j_2 \cdots j_N}(t) = \beta_{j_1 j_2 \cdots j_N}(\tau_{j_1 j_2 \cdots j_N}) \tag{4}$$

where $\tau_{j_1 j_2 \cdots j_N}$ is the most recent time when the path $\phi_{j_1 j_2 \cdots j_N}$ has been chosen, and $\beta_{j_1 j_2 \cdots j_N}(\tau_{j_1 j_2 \cdots j_N})$ is the reward from the environment at $\tau_{j_1 j_2 \cdots j_N}$.

Definition 2 Let $v_{i_1 i_2 \cdots i_{s-1}}(t) = (v_{i_1 i_2 \cdots i_{s-1} 1}(t), v_{i_1 i_2 \cdots i_{s-1} 2}(t), \ldots, v_{i_1 i_2 \cdots i_{s-1} r}(t))'$ be the recent reward vector corresponding to the sth level learning automaton $Z_{i_1 i_2 \cdots i_{s-1}}$ ($s = 1, 2, \ldots, N$).

[3] $E\{\bullet\}$ denotes the mathematical expectation of $\bullet$.

Here, each of the components of $v_{i_1 i_2 \cdots i_{s-1}}(t)$ is constructed as follows:

i) At the Nth level,

$$v_{i_1 i_2 \cdots i_N}(t) = u_{i_1 i_2 \cdots i_N}(t) \tag{5}$$

ii) At the sth level ($1 \leq s \leq N-1$),

$$v_{i_1 i_2 \cdots i_s}(t) = \max\{v_{i_1 i_2 \cdots i_s 1}(t), v_{i_1 i_2 \cdots i_s 2}(t), \ldots, v_{i_1 i_2 \cdots i_s r}(t)\} \tag{6}$$

As in [20], we also assume that the following condition holds for all $i_1, i_2, \ldots, i_s$ ($i_q = 1, 2, \ldots, r;\ q = 1, 2, \ldots, s$ ($s = 1, 2, \ldots, N$)):

$$q_{min} \leq p_{i_1 i_2 \cdots i_s}(t) \leq q_{max} \tag{7}$$

where q_{min} and q_{max} satisfy the inequalities $0 < q_{min} < q_{max} < 1$ and $q_{max} = 1 - (r-1) q_{min}$.

Next we shall propose a new learning algorithm which is an extended form of Simha and Kurose [20] relative reward strength algorithm.

Learning Algorithm

Assume that the path $\phi(t) = \phi_{j_1 j_2 \cdots j_N}$ has been chosen at time t and (corresponding to the output from the hierarchical structure automata system) the environmental response $u_{j_1 j_2 \cdots j_N}$ has been given. Then, the action probabilities $p_{j_1 j_2 \cdots j_{s-1} i_s}(t)$ ($i_s = 1, 2, \ldots, r$) of each automaton $Z_{j_1 j_2 \cdots j_{s-1}}$ ($s = 1, 2, \ldots, N$) connected to the path being chosen are updated by the following equation:

$$\begin{aligned} p_{j_1 j_2 \cdots j_{s-1} i_s}(t+1) &= p_{j_1 j_2 \cdots j_{s-1} i_s}(t) \\ &\quad + \lambda_{j_1 j_2 \cdots j_{s-1}}(t) \Delta p_{j_1 j_2 \cdots j_{s-1} i_s}(t) \end{aligned} \tag{8}$$

where $\Delta p_{j_1 j_2 \cdots j_{s-1} i_s}(t)$ is calculated by

$$\Delta p_{j_1 j_2 \cdots j_{s-1} i_s}(t) = \begin{cases} v_{j_1 j_2 \cdots j_{s-1} i_s}(t) - \dfrac{1}{|A_s(t)|} \displaystyle\sum_{l_s \in A_s(t)} v_{j_1 j_2 \cdots j_{s-1} l_s}(t), & {}^\forall i_s \in A_s(t) \\ 0, & i_s \notin A_s(t) \end{cases} \tag{9}$$

Here, the set $A_s(t)$ is constructed as follows:

a) Line up $v_{j_1 j_2 \cdots j_{s-1} i_s}(t)$ in descending order.

b) Set $D_{j_1 j_2 \cdots j_{s-1}} = \{k_s | v_{j_1 j_2 \cdots j_{s-1} k_s}(t) = \max_{i_s}\{v_{j_1 j_2 \cdots j_{s-1} i_s}(t)\}\}$.

c) Repeat the following procedure for i_s ($i_s \notin D_{j_1 j_2 \cdots j_{s-1}}$) in descending order of $v_{j_1 j_2 \cdots j_{s-1} i_s}(t)$:
If the inequality $p_{j_1 j_2 \cdots j_{s-1} i_s}(t+1) > q_{min}$ can be satisfied as a result of calculation by (8) and (9), then set $D_{j_1 j_2 \cdots j_{s-1}} = D_{j_1 j_2 \cdots j_{s-1}} \bigcup \{i_s\}$.

d) Set $A_s(t) = D_{j_1 j_2 \cdots j_{s-1}}$.

Remark 2 $\lambda_{j_1 j_2 \cdots j_{s-1}}(t)$ is the stepsize parameter at time t.

Remark 3 In the proposed algorithm, the change in the action probability of each actuated automaton in the hierarchy is propotional to the difference between the corresponding component of the recent reward vector and the average recent reward over all actions in the actuated automaton.

Remark 4 The action probabilities of each automaton which is not on the selected path are not changed.

Remark 5 As we have already mentioned, the proposed algorithm has been obtained by considering the extension of the relative reward strength algorithm proposed by Simha and Kurose [20]. This means that the proposed algorithm coincides with the original one when $N = 1$. Several computer simulation results [20] showed that the relative reward strength algorithm (used in the single automaton) outperforms the SL_{R-I} scheme under the nonstationary random environment with high noise.

5. Convergence Theorem

In this section, we shall derive a convergence theorem concerning the learning performance of our proposed algorithm. First, we can obtain the following lemma by paying attention to the components of the reward vector $v_{j_1^* j_2^* \cdots j_{s-1}^*}(t) = (v_{j_1^* j_2^* \cdots j_{s-1}^* 1}(t), v_{j_1^* j_2^* \cdots j_{s-1}^* 2}(t), \ldots, v_{j_1^* j_2^* \cdots j_{s-1}^* r}(t))'$ $(s = 1, 2, \ldots, N)$ corresponding to the optimal path $\phi_{j_1^* j_2^* \cdots j_N^*}$:

Lemma 1 Suppose that each component of the current reward vector is given by (5) and (6). Then, the following inequality concerning the learning automaton $Z_{j_1^* j_2^* \cdots j_{s-1}^*}$ $(s = 1, 2, \ldots, N)$ holds.

$$E\{v_{j_1^* j_2^* \cdots j_s^*}(t)\} > E\{v_{j_1^* j_2^* \cdots j_{s-1}^* i_s}(t)\} \tag{10}$$

where, $i_s = 1, 2, \ldots, r$, $i_s \neq j_s^*$.

By taking advantage of the lemma 1, we can obtain the following theorem concerning the convergence to the optimal path $\phi_{j_1^* j_2^* \cdots j_N^*}$:

Theorem 1 Assume that the condition (7) and the conditions given in the lemma hold. Further, let $\lambda_{j_1 j_2 \cdots j_{s-1}}(t)$ be a sequence of real numbers such that

$$\lambda_{j_1 j_2 \cdots j_{s-1}}(t) > 0, \quad \sum_{t=1}^{\infty} \lambda_{j_1 j_2 \cdots j_{s-1}}(t) = \infty, \quad \sum_{t=1}^{\infty} \lambda^2_{j_1 j_2 \cdots j_{s-1}}(t) < \infty \tag{11}$$

Then, the path probability $\pi_{j_1^* j_2^* \cdots j_N^*}(t)$ that the hierarchical structure automata system chooses the optimal path $\phi_{j_1^* j_2^* \cdots j_N^*}$ at time t converges almost surely to $(q_{max})^N$.

Remark 6 Due to space limitation, we omit the proofs of Lemma 1 and Theorem 1. Interested readers are kindlly asked to read the paper[27].

6. Computer Simulation Results

In order to investigate whether the proposed algorithm can be successfully utilized in the nonstationary S-model environments, we carried out many computer simulations.

We shall show one of the computer simulation results concerning the learning behaviors of the HSLA. Before going into details concerning the computer simulation, we shall briefly explain the hierarchical automata system, nonstationary environments, etc.

A. Hierarchical Structure Automata Model

The hierarchical structure automata model is characterized by the following:

1) Number of the levels of the hierarchical structure learning automata: 11
2) Number of the actions of each automaton in the hierarchy: 2

Table 1 VALUES OF THE COEFFICIENTS $a_{i_1 i_2 \cdots i_N}$ & $b_{i_1 i_2 \cdots i_N}$

path	a	b
Optimal Path $\phi_{11111221121}$	0.88	0.05
Second optimal path $\phi_{11111112122}$	0.70	0.08

3) Total number of paths: 2048
4) Optimal path: $\phi_{11111221121}$

B. Nonstationary Environment

We have considered the following nonstationary environment:

1) The environmental reward $\beta(t)$ at time t corrsponding to the output $\phi(t) = \phi_{i_1 i_2 \cdots i_N}$ from the hierarchy is characterized by the following equation:

$$\beta(t) = \beta_{i_1 i_2 \cdots i_N}(t) + e_{i_1 i_2 \cdots i_N} \xi \tag{12}$$

where ξ is the random variable with the uniform probability density function in the closed interval [-0.5, 0.5] and $\beta_{i_1 i_2 \cdots i_N}(t)$ $= a_{i_1 i_2 \cdots i_N} + b_{i_1 i_2 \cdots i_N} \sin(c_{i_1 i_2 \cdots i_N} \pi t + d_{i_1 i_2 \cdots i_N})$. Here, $a_{i_1 i_2 \cdots i_N}$, $b_{i_1 i_2 \cdots i_N}$, $c_{i_1 i_2 \cdots i_N}$, $d_{i_1 i_2 \cdots i_N}$ and $e_{i_1 i_2 \cdots i_N}$ are the positive scalors whose values have been chosen in such a way that the inequality $0 \leq \beta(t) \leq 1$ holds for all t.

2) In the simulation, we utilized the following particular combinations concerning the parameter values.

$a_{i_1 i_2 \cdots i_N}, b_{i_1 i_2 \cdots i_N}$: The values of $a_{i_1 i_2 \cdots i_N}$ and $b_{i_1 i_2 \cdots i_N}$ corresponding to the optimal path and the second optimal path are given in Table 1. The values of the other $a_{i_1 i_2 \cdots i_N}$ have been chosen by using the random variable with the uniform probability density function in the closed interval [0.1, 0.5]. On the other hand, the values of the other $b_{i_1 i_2 \cdots i_N}$ have been chosen to be equal to the same value 0.1.

$c_{i_1 i_2 \cdots i_N}$: The value of $c_{i_1 i_2 \cdots i_N}$ has been chosen by using the random variable with the uniform probability density function in the closed interval [0.3, 0.8].

$d_{i_1 i_2 \cdots i_N}$: 0.5 for all $i_1, i_2, \ldots, i_N$.

$e_{i_1 i_2 \cdots i_N}$: 0.03 for all $i_1, i_2, \ldots, i_N$.

C. Parameters q_{min} & q_{max}

As the parameters q_{min} & q_{max}, we have used the following values:

$q_{min} = 0.002$

$q_{max} = 0.998$

We have used the same value of the parameter $\lambda_{i_1 i_2 \cdots i_{s-1}}$ $(s = 1, 2, \cdots, N)$ from the top level of the hierarchy to the bottom level.

Table 2 shows the average number of iteration and the probability that the proposed algorithm has succeeded in finding the optimal path. In each computer experiment, we have carried out 30 simulations. In order to compare our proposed algorithm with the familiar learning algorithm by Thathachar and Ramakrishnan [8], we have also carried out computer simulations using their algorithm by keeping the same experimental

Table 2 SIMULATION RESULTS (OUR ALGORITHM)

Step size parameters	Average Number of Iterations	Percentage of correct convergences(%)
0.01	2234.0	100
0.02	1188.2	100
0.03	851.4	100
0.04	8640.5	100
0.05	11944.1	100

Table 3 SIMULATION RESULTS (T&R ALGORITHM)

Step size parameters	Average Number of Iterations	Percentage of correct convergences(%)
0.00025	127132.4	100
0.00026	131674.1	100
0.00027	126888.2	91
0.00028	126098.2	100
0.00029	113611.0	91
0.0025	11398.4	37
0.0026	10978.6	23
0.0027	12277.0	30
0.0028	11268.2	30
0.0029	10433.4	23

conditions. Table 3 shows the computer simulation results. From these results, we may confirm the effectiveness of our proposed algorithm.

7. A Proposal of a Learning Algorithm of HSLA Operating in the Nonstationary Multiteacher Environment

We have recently succeeded in constructing a learning algorithm of HSLA operating in the nonstationary multiteacher environment (as shown in Fig.3). Due to limitation of space, we don't go into details. Interested readers are kindly asked to attend our presentations.

8. Conclusions

In this paper, we have extended the relative reward strength algorithm of Simha and Kurose [20] in order that it can be used in the HSLA model. We have indicated that the proposed algorithm ensures convergence to the optimal action w.p.1 under the certain type of nonstationary S-model environment. Future research is needed to investigate the learning behaviors of the hierarchical structure automata under various types of the nonstationary environments.

Acknowledgement

The authors would like to thank FOST who has given them financial support.

References

(1) M.L. Tsetlin, "On the behavior of finite automata in random media", Avtomatika i Telemekhanika, vol.22-10, pp.1345-1354 1961.
(2) V.I. Varshavskii and I.P. Vorontsova, "On the behavior of stochastic automata with variable structure", Automation and Remote Control, vol.24, pp.327-333, 1963.
(3) S. Lakshmivarahan, Learning Algorithms Theory and Applications, Springer-Verlag, 1981.
(4) K.S. Narendra and M.A.L. Thathachar, "Learning automata - A survey", IEEE Trans. Syst., Man, Cybern., vol.4, pp.323-334, 1974.
(5) N. Baba, New Topics in Learning Automata Theory and Applications, Springer-Verlag, 1985.
(6) K.S. Narendra and M.A.L. Thathachar, Learning Automata: An Introduction, Englewood Cliffs, NJ: Prentice-Hall, 1989.
(7) A.S. Poznyak and K. Najim, Learning Automata and Stochastic Optimization, Springer-Verlag, 1997.
(8) M.A.L. Thathachar and K.R. Ramakrishnan, "A hierarchical system of learning automata", IEEE Trans. Syst., Man, Cybern., vol. SMC-11, pp.236-241, 1981.
(9) K.R. Ramakrishnan, Hierarchical Systems and Cooperative Games of Learning Automata, Ph.D. Thesis, Indian Institute of Science, Bangalore, 1982.
(10) N. Baba, "Learning behaviors of hierarchical structure stochastic automata operating in a general multiteacher environment", IEEE Trans. Syst., Man, Cybern., vol. SMC-15, pp.585-587, 1985.
(11) M.A.L. Thathachar and V.V. Phansalkar, "Learning the global maximum with parameterized learning automata", IEEE Trans. Neural Networks, vol.6, pp.398-406, 1995.
(12) M.A.L. Thathachar and V.V. Phansalkar, "Convergence of teams and hierarchies of learning automata in connectionist systems", IEEE Trans. Syst., Man, Cybern., vol.25, pp.1459-1469,1995.
(13) M.A.L. Thathachar and P.S. Sastry, "A class of rapidly converging algorithms for learning automata", Proceedings of the IEEE International Conference on Cybernetics and Society, Bombay, India, pp.602-606, 1984.
(14) M.A.L. Thathachar and P.S. Sastry, "A new approach to the design of reinforcement schemes of learning automata", IEEE Trans.SMC, vol. SMC-15, pp.168-175, 1985.
(15) M.A.L. Thathachar and P.S. Sastry, "A hierarchical system of learning automata that can learn the globally optimal path", Information Sciences, vol.42, pp.143-166, 1987.
(16) B.J. Oommen and J.K. Lanctot, "Discretized pursuit learning automata", IEEE Trans. Syst., Man, Cybern., vol. SMC-20, pp.931-938, 1990.
(17) G.I. Papadimitriou, "A new approach to the design of reinforcement schemes for learning automata: stochastic estimator learning algorithms", IEEE Trans. Knowledge and Data Engineering, vol.6, pp.649-654, 1994.
(18) G.I. Papadimitriou, "Hierarchical discretized pursuit nonlinear learning automata with rapid convergence and high accuracy", IEEE Trans. Knowledge and Data Engineering, vol.6, pp.654-659, 1994.
(19) B.J. Oommen and M. Agache, "Continuous and discretized pursuit learning scheme: various algorithms and their comparison", IEEE Trans. Syst., Man, Cybern. B, vol.31, pp.277-287, 2001.
(20) R. Simha and J.F. Kurose, "Relative reward strength algorithms for learning automata", IEEE Trans. Syst., Man, Cybern., vol.19, pp.388-398, 1989.
(21) M.A.L. Thathachar and M.T. Arvind, "Parallel algorithms for modules of learning automata", IEEE Trans. Syst., Man, Cybern., vol.28, pp.24-33, 1998.
(22) P.R. Srikantakumar and K.S. Narendra, "A learning model for routing in telephone networks", SIAM J. Control and Optimization, vol.20, pp.34-57, 1982.
(23) N. Baba and Y. Sawaragi, "On the learning behavior of stochastic automata under a nonstationary random environment", IEEE Trans. Syst., Man, Cybern., vol. SMC-5, pp.273-275, 1975.
(24) N. Baba, "On the learning behaviors of variable-structure stochastic automaton in the general N-teacher environment", IEEE Trans. Syst., Man, Cybern., vol. SMC-13, pp.224-231, 1983.
(25) N. Baba and Y. Mogami, "Learning behaviors of hierarchical structure stochastic automata in a nonstationary multi-teacher environment", International Journal of Systems Science, vol.19, pp.1345-1350, 1988.
(26) X. Zeng, J. Zhou, C. Vasseur, "A strategy for controlling nonlinear systems using a learning automaton", Automatica, vol.36, pp.1517-1524, 2000.
(27) N. Baba and Y. Mogami, "A New Learning Algorithm for the Hierarchical Structure Learning Automata Operating in the Nonstationary S-model Random Environment", IEEE Trans. SMC, Part B, Vol.36, No.6, December, 2002 (to be published).

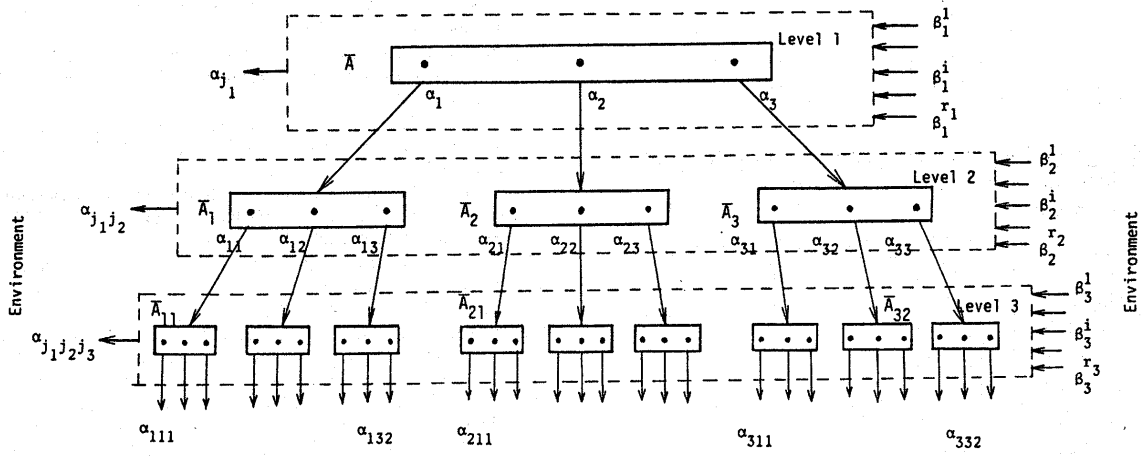

Fig. 3 HSLA operating in the multiteacher environment