The Acquisition of Space Search Procedure Depending on Agent Structure

A.Ogawa & T.Omori
Graduate School of Engineering, Hokkaido University,
Sapporo, 065-8628, Japan
akitoshi@complex.eng.hokudai.ac.jp
omori@complex.eng.hokudai.ac.jp

ABSTRACT

In this paper, we propose the architecture that searches and acquires calculation procedure. In this idea, the calculation procedure of intellectual system is realized by a set of functional parts and an attention sequence combining them. To verify this idea, we see the development of agent behavior after a change of its subfunctional parts. At first, an agent acquires Q-learning procedure using given parts. When an environmental map representing subfunction is added to the agent, its behavior changes to more effective search using the map for prediction.

Keywords: procedure acquisition, functional parts, attention search

1. INTRODUCTION

In a study of intellectual information processing model such as machine learning, the concrete design of a processing system for each task is as important an issue as the abstract level modeling of data representation and its operation. However, in the model research, the concrete processing procedure required to realize the model and to perform peripheral computation including preprocessing, central calculation, execution of calculation result and conducting control, was entrusted to a human who knows the task in advance. Therefore, even if the model of an intelligence system is powerful, a human being must intervene and implement the system separately. When the task changed, the model had to be re-implemented.

Contrarily, humans can perform tasks that require essentially the same calculation with different appearance and interface, detecting the identity of their processes and adding the function that absorbs the external difference. They can also solve a task that has similar appearance with different central calculation by noticing the difference and replacing the calculation with one suitable for the task. Behind this ability, we should assume architecture of intelligence in which there are many functional parts for intellectual calculation and they are selected and combined as the need arises. In this paper, we propose the combination mechanism of functional parts using an attention sequence as the architecture that looks for a calculation procedure to realize the intellectual behavior.

> Calculation procedure of intellectual system is realized by a set of functional parts and an attention sequence that combines them. The attention sequence alters the calculation process from input to output by changing the combination of functional parts according to environment and situation. The mechanism that selects combination of functional parts depending on situation realizes flexible procedure search system for new problems.

To verify this idea, we choose a navigation problem as a benchmark task, and by a computer simulation show that more effective path search procedure is acquired by the addition of functional parts. We prepare a grid world for the navigation simulation to simplify the argument. We assume an agent that searches for the shortest path to the goal, moving through each grid. For the agent structure, we use a neural network with activity control that can represent dynamica change of the processing system naturally.

We show computer simulation results that indicate the possibility of procedure acquisition including learning ability. First, an agent acquires Q-learning procedure using the given parts. When the agent is given the functional parts for the environmental map representation, it acquires prediction-based navigation strategy more effective than the previous one.

2. THE ARCHITECTURE OF PROCEDURE SEARCH

In our idea, a model has many functional parts that are specialized for different subfunctions, and is controlled by the system which combines those functional parts depending on a task, and performs an operation based on internal knowledge. We consider this process the procedure.

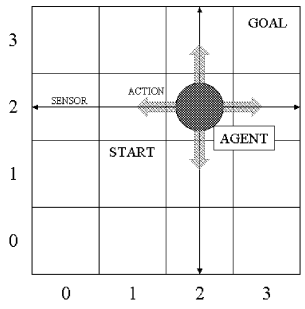

Figure 1: The world for procedure acquisition

Omori and Mochizuki[3] proposed a method that uses sequence of attention for realization of this internal procedure. In that method, the Attention System selects a set of neural network modules which are required for instantaneous processing and their operation form a processing circuit. Furthermore, the attention changes sequentially forming the dynamically changing neural network. The result is the realizatino of the sequential operation of memorized knowledge.

In this paper, we adopt a modified version of the method to realize a procedure. In our method, the attention directly switches connections between the neural parts to form a processing circuit.

We use the genetic algorithm (GA) for the searching method of the parts combination. Fig.2 shows basic structure of our agent for the procedure search.

As a result, the agent consists of some functional parts and their connections. The range of realizable procedure functions change depending on what functional parts are prepared in the agent.

3.NAVIGATION PROBLEM

The task in this paper is a goal search problem in the 4x4 grid world (Fig.1). The agent receives the distance to the wall in the four directions as a sensory input, and has internal states corresponding to each grid.

Start and goal points are located at (1,1) and (3,3) in Fig.1 respectively. When the agent reaches the goal, it receives reward and returns to the starting point. Should the agent collide with the wall, it receives no reward and goes back to the starting point. The task is to maximize the reward. In short, the agent must discover the shortest path to the goal quickly.

4. AGENT STRUCTURE

For comparison, we prepare two agents, a basic agent and an evolved agent. The Basic agent's functional parts are sensory input, place recognition, random action generation, learned action generation, sufficient Q-table memory for Q-learning and a maximum value search circuit. The evolved agent has all of the avobe, as well as an internal map of the environment. Both agents also have the same attention system.

4.1 Basic Agent

The basic agent consists of the following functional parts.

- State recognition
- Action selection
- Reinforcement learning
- Attention System

4.1.1 The Part of State Recognition

Distance sensor input ***RS*** is normalized. the SP-layer receives it and works as an input buffer.

$$SP_i = \frac{RS_i}{\sqrt{\sum_i RS_i^2}}$$

Each cell of the SS-layer represents the internal state corresponding to the place in the environment. Competition between cells limits the number of cells firing at any one time to one.

$$\begin{aligned} ss_j &= a_1 \sum_i SW_{ij} SP_i \qquad (1) \\ w &= \arg\max_j ss_j \\ SS_j &= \begin{cases} 0 & : j \neq w \\ 1 & : j = w \end{cases} \end{aligned}$$

The variable a_1 is the attention value over the connection SW_{ij} from the SP-layer to the SS-layer. It takes a value of zero or one to control usage of the connection SW_{ij}.

4.1.2 The Part of Action Selection

Input to the MP-layer is a vector $(\cos\theta, \sin\theta)$ that is generated by random action generator , where θ is a random value between 0 and 2π. It generates random action when proper attention is set to the neural circuit.

Each cell of the MS-layer represents the discrete motor direction. Only one cell is allowd to fire at any one time.

$$ms_u = a_4(\sum_r MW_{ru} MP_r^{in}) + a_5 Q_u \qquad (2)$$

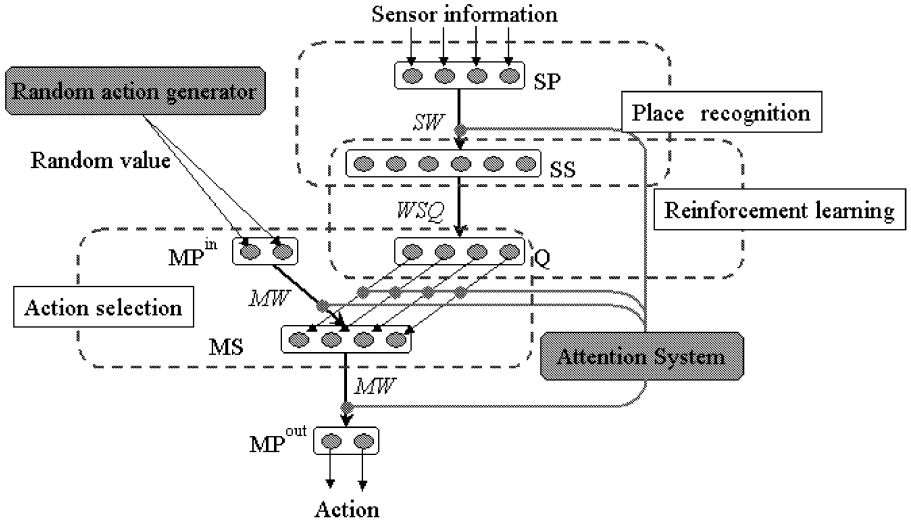

Figure 2: The structure of a basic agent

$$w' = \arg\max_u ms_u$$

$$MS_u = \begin{cases} 0 & : u \neq w' \\ 1 & : u = w' \end{cases}$$

After the firing of an MS-layer cell, its corresponding input pattern MW_{ru} is recollected at MP^{out}-layer, and it is outputted.

$$MP_r^{out} = a_8 \sum_u MW_{ru} MS_u$$

The variable a_4 is the attention value over the connection MW_{ru} from the MP-layer to the MS-layer, and the a_5 is the attention value over the connection from the Q-layer to the MS-layer. Both variables take the value of zero or one, and control the usage of the corresponding term in each equation. The a_8 works as an action trigger.

4.1.3 The Part of Reinforcement Learning

After the place representing cell SS_j fires, the Q-layer receives the action-value (Q-value) of the place and works as a buffer for this value. Because there is a one to one correspondence between cells in the Q-layer and the MS-layer, each of Q-layer cell activity represents the Q-value of each action k at the place j.

$$Q_k = \sum_j WSQ_{jk} SS_j$$

By inputting this value to the MS-layer, the agent can decide its action based on the Q-value.

The connection WSQ_{jk} is equivalent to the Q-value table in Q-learning. Learning of WSQ_{jk} is performed by the following equation.

$$WSQ_{w_{t-1},w'} = \alpha(r + \gamma(\max_b WSQ_{w_t,b} - WSQ_{w_{t-1},w'}))$$

Here α is the learning rate, and γ is the discount rate. This is a typical Q-learning equation. Although acquisition of the learning rule itself is only one of our final targets, we give it a priori in this study.

4.1.4 Attention System

The Attention System can control the topology of a neural network dynamically by changing the attention vector. The attention vector can be considered the equivalent of microcode in CPU. Using the attention vector, the programming of a neural network might be possible by controlling the network structure. However, so far as the attention vector designates the structure at one moment, it can not describe even easy algorithms. So, we assume an attention vector sequence (AVS) that consists of two or more ordered attention vectors. Each AVS element is given to the network sequentially, and the network operates until it converges to a stable state. Then, the next element is given to the network.

$$\boldsymbol{avs} = (\boldsymbol{av_0}, \boldsymbol{av_1}, \boldsymbol{av_2}, \ldots, \boldsymbol{av_n})$$

The attention vector of basic agent consists of the following four elements.

- a_1:switching of inhibition on input from SP-layer to SS-layer
- a_4:switching of inhibition on input from MP-layer to MS-layer

- a_5:switching of inhibition on input from Q-layer to MS-layer
- a_8:switching of inhibition on input from MS-layer to MP-layer (triggering of action).

4.2 Evolved Agent

In addition to the functional parts of the basic agent, the evolved agent has the internal model that represents the environmental map. Our interest resides in the change of agent behavior after the addition of this type of map.

4.2.1 State Recognition Part

In addition to the input from sensor, the SS-layer of the evolved agent receives input from the T-layer and the Q-layer. Equation (1) is changed as follows.

$$\begin{aligned} ss_j &= a_1 \sum_i SW_{ij} SP_i + a_2 \sum_l WTS_{lj} T_l \\ &\quad + a_3 \sum_k WSQ_{jk} Q_k \end{aligned}$$

The 2nd term is the input from the T-layer that represents the place-action to place relation of environment. The value of a_2 and a_3 represent attention over each input.

4.2.2 Action Selection Part

The MS-layer of the evolved agent also receives input from the A-layer. Equation (2) is modified to include the input from the A-layer that represents the place-place to action relation.

$$\begin{aligned} ms_u &= a_4 \sum_r MW_{ru} MP_r^{in} + a_5 Q_u \\ &\quad + a_6 \sum_v WAM_{vu} A_v \end{aligned}$$

The value a_6 represents attention over the input.

4.2.3 Environment Model Learning

The T-layer represents the next state from the combination of the current internal state and the action. As the state corresponds to a place in the map, the T-layer represents the state transition relation of the environment. The number of T-layer cells is the product of the number of SS-layer cells and the number of MS-layer cells. WST_{jl} and WMT_{ul} are set by the following equations, where N is the number of SS-layer cells.

$$\begin{aligned} WST_{jl} &= \begin{cases} 1 & : l = uN + j \\ 0 & : others \end{cases} \\ WMT_{ul} &= \begin{cases} 0 & : l = um + 0 \ldots N - 1 \\ -1 & : others \end{cases} \end{aligned}$$

The value of a T-layer cell is calculated by the following equation.

$$\begin{aligned} T_l &= \phi(\sum_j WST_{jl} SS_j + \sum_u WMT_{ul} MS_u) \\ \phi(x) &= \begin{cases} 0 & : x \leq 0 \\ x & : 0 < x < 1 \\ 1 & : x \geq 1 \end{cases} \end{aligned}$$

The connection WTS_{lj} between the T-layer and the SS-layer represents state transition, (state, action) $\rightarrow$ state. The T-layer is used to predict the next state based on the next action. Learning of WTS_{lj} is performed by the following equation,

$$WTS_{lj} = WTS_{lj} + \alpha(T_l - WTS_{lj}) SS_j$$

where all the initial value of WTS_{lj} is 0.

The A-layer represents the action that bridges internal state of time t and $t-1$. The number of A-layer cells is equal to the square of the number of SS-layer cells. We assume the SS'-layer represents the activity of the SS-layer on time $t-1$. Connections WSA_{jv} and $WS'A_{j'v}$ are set by the following equations, where N is the number of SS-layer cells.

$$\begin{aligned} WSA_{jv} &= \begin{cases} 1 & : v = j(N-1) + 1, \ldots, N-1 \\ 0 & : others \end{cases} \\ WS'A_{j'v} &= \begin{cases} 1 & : j' + j(N-1) \quad (j' \neq j) \\ 0 & : others \end{cases} \end{aligned}$$

The value of an A-layer cell is calculated by the following equation,

$$A_v = (\sum_j WSA_{jv} SS_j)(\sum_u WS'T_{j'v} SS'_{j'})$$

The connection WAM_{vu} lies between the A-layer and the MS-layer,and represents the directional relation between the places. Learning of WAM_{vu} is performed by following equation,

$$WAM_{vu} = WAM_{vu} + \alpha(A_v - WAM_{vu}) MS_u$$

where all the initial value is set to zero.

4.2.4 Reinforcement Learning Part

Special attention value a_7 is added to the Q-layer of an evolved agent. If $a_7 = 0$, the Q-layer works as same as that of the basic agent. If $a_7 = 1$, all the cells of the Q-layer output value 1.

If $a_7 = 1$ and the SS-layer receives input from the Q-layer and the T-layer, the total Q-value at each place is represented at corresponding cell in the SS-layer by the input from the Q-layer. The input from the T-layer restricts the firing cell in the SS-layer to

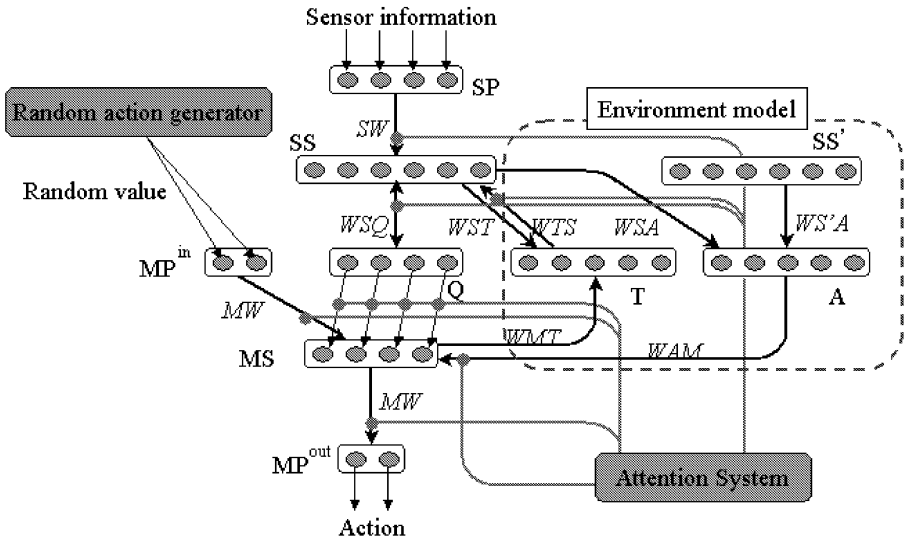

Figure 3: The structure of an evolved agent

those reachable in a single step. With these inputs, the cell that corresponds to the place with the maximum Q-value fires, due to the embedded lateral inhibition property.

4.2.5 Attention System

The following four elements are added to the attention vector of evolved agent.

- a_2:switching of inhibition on input from T-layer to SS-layer
- a_3:switching of inhibition on input from Q-layer to SS-layer
- a_6:switching of inhibition on input from A-layer to MS-layer
- a_7:switching of firing all Q-layer cells

5. SIMULATION

In our model, the procedure acquisition is realized by the acquisition of AVS. Because AVS is a bit sequence of $\{0, 1\}$, searching is possible by the use of genetic algorithm (GA).

5.1 Preparation

The length of AVS is set to two, and the population size is set to 50. All genes are initialized randomly.

One trial ends when an agent discovers the shortest path. Fitness O is the number of necessary moves in each trial. When an agent reaches the goal three times consecutively in the minimum number of steps, we conclude that the agent has found the shortest path. It is unreliable to decide fitness from one trial, because the action of agent contains random factor. Accordingly, the following scale is used.

$$f(O) = 1000 \sum_{i=1}^{10} O_i^{-1}$$

To reduce the influence of random factor, the sum of O^{-1} for ten trials per individual is calculated.

Half the agents with higher fitness are used for the next generation production. Crossover pairs are decided in the order of 1st and 2nd, 2nd and 3rd ... and so on based on their fitness value. One-point crossover is used, and crossover point is chosen at random. The mutation rate is 0.05.

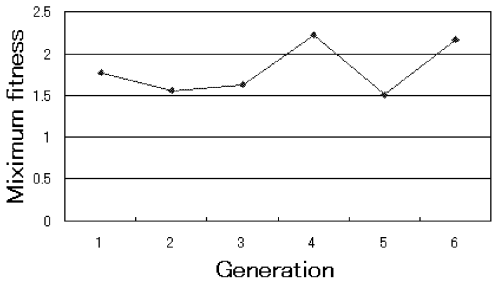

Figure 4: The fitness of the best adapted basic agent for each generation.

5.2 Result

5.2.1 Basic Agent

The maximum fitness of the basic agent in each

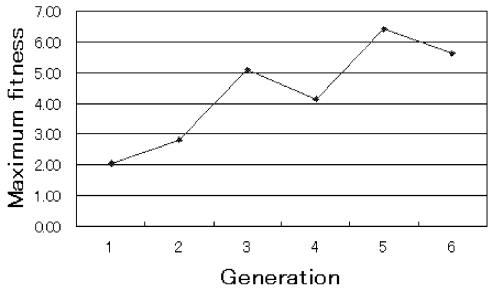

Figure 5: The fitness of the best adapted evolved agent for each generation. The fitness is improved at the 3rd generation, by the acquisition of prediction-based action.

generation is shown in Fig.4.

The following two AVSs are typical ones that the basic agent acquired.

$$\boldsymbol{avs} = ((1,0,1,0),(0,0,1,1)) \quad (3)$$

$$\boldsymbol{avs} = ((1,0,1,1),(0,0,1,1)) \quad (4)$$

The basic agent acquired AVS (3) at the 6th generation. The first attention vector recognizes the current state. According to the Q-value, the action is selected and conducted by the 2nd attention vector. It is a well-known greedy strategy.

The basic agent acquired AVS (4) at the 3rd generation. The current state is recognized and action is conducted according to the Q-value based on the 1st attention vector. The first step action is repeated based on the 2nd attention vector. The agent accelerates reaching the goal by repeating the same action.

5.2.2 Evolved Agent

The maximum fitness of the evolved agent in each generation is shown in Fig.5.

The evolved agent acquired the following AVS at 5th generation.

$$\boldsymbol{avs}_B = ((1,0,0,0,1,0,1,0),(0,0,1,0,1,1,0,1))$$

An embedded competition process in the 1st attention vector finds the next place with the maximum total sum of Q-value within the range of a single step. The agent moves to the place by the 2nd attention vector. This is a prediction-based behavior that makes use of the environmental map.

6. CONCLUTION

In this paper, we propose an architecture that searches and aqcuires calculation procedure. We show that a different path search procedure was acquired, when the functional parts are different. The simulation results show that the agent has the ability to acquire an effective procedure if it has the functional parts that are effective to path search problem. This simulation was not sufficient to show that the procedure effectiveness is dominated by the functional parts effectiveness to the task. For that, we need to perform an additional simulation.

References

[1] Atkeson, C. G., Santamaria, J. C.: A Comparison of Direct and Model-Based Reinforcement Learning, International Conference on Robotics and Automation, 1997.

[2] Mizutani, K.,Omori, T.: On-line Map Formation and Path Planning for Mobile Robot by Associative Memory with Controllable Attention, Proc. of IJCNN'99, 1999.

[3] Omori, T., Mochizuki, A., Mizutani, K., Nishizaki. M.: Emergence of symbolic behavior from brain like memory with dynamic attention, Neural Networks , Vol.12, 1157-1172, 1999.

[4] Sutton, R. S., Barto, A. G.: Reinforcement Learning: An Introduction, MIT press, 1998.

[5] Trullier, O., Meyer, J.A.: Animat navigation using a cognitive graph, Biological Cybernetics, Vol.83, no.3, pp271-285, Springer, 2000.