Hierarchical Reinforcement Learning Using A Modular Fuzzy Model for Multi-Agent Problem

Toshihiko Watanabe and Yoshiya Takahashi, Member, IEEE

Abstract— Reinforcement learning is a promising approach to realize intelligent agent such as autonomous mobile robots. In order to apply the reinforcement learning to actual sized problem, the “curse of dimensionality” problem in partition of sensory states should be avoided maintaining computational efficiency. The paper describes a hierarchical modular reinforcement learning that Profit Sharing learning algorithm is combined with Q-Learning reinforcement learning algorithm hierarchically in multi-agent pursuit environment. As the model structure for such the huge problem, we propose a modular fuzzy model extending SIRMs architecture. Through numerical experiments, we found that the proposed method has good convergence property of learning compared with the conventional algorithms.

I. INTRODUCTION

Reinforcement learning[1-5] is a promising approach to realize intelligent agent such as autonomous mobile robots. However there exist a lot of problems compared with the other learning techniques such as Neural Networks in order to apply reinforcement learning to actual applications. One of the main problems of reinforcement learning application of actual sized problem is “curse of dimensionality” problem in partition of multi-inputs sensory states. High dimension of input leads to huge number of rules in the reinforcement learning application. It should be avoided maintaining computational efficiency for actual applications. Multi-agent problem such as the pursuit problem[6,7] is difficult for reinforcement learning computation in terms of huge dimensionality. As the other related problem, learning of complex task is not easy essentially because the reinforcement learning is based only upon rewards derived from the environment.

In order to deal with these problems, several effective approaches are studied. For relaxation of task complexity, several types of hierarchical reinforcement learning have been proposed to apply actual applications[8,9]. To avoid the curse of dimensionality, there exists modular hierarchical learning[10,11] that construct the learning model as the combination of subspaces. Adaptive segmentation[12,13] for constructing the learning model validly corresponding to the environment is also studied. However more effective technique of different approach is also necessary in order to apply reinforcement learning to actual sized problems.

In this paper, we focus on the well-known pursuit problem and propose a hierarchical modular reinforcement learning that Profit Sharing learning algorithm is combined with Q-Learning reinforcement learning algorithm hierarchically in multi-agent environment. As the model structure for such the huge problem, we propose a modular fuzzy model extending SIRMs architecture[14][15]. Through numerical experiments, we show the effectiveness of the proposed algorithm compared with the conventional algorithms.

II. PURSUIT PROBLEM AS MULTI-AGENT ENVIRONMENT

The pursuit problem is well known and studied as typical benchmark problem in Distributed Artificial Intelligence research field[6]. It is multi-agent based problem that hunter agents act collaboratively to capture prey agent. For actual computer simulations or mobile robot applications, it is indispensable to avoid huge memory consumption for the state space, i.e. “curse of dimensionality”, and to improve slow learning speed caused by its sparsity(e.g. acquired Q-value through reinforcement learning). Figure 1 shows the 4-agent pursuit problem in 7x7 grids field. In the problem, all agents behave simultaneously to move upward, downward, rightward, leftward in one gird, or to stay. Collision of the agents is settled randomly because one gird allows only one agent to stay. The objective of the simulation is to surround the prey agent by the hunter agents as shown in Fig.2.

Fig.1. 4-Agent Pursuit Problem (7x7)
The hunter agents can utilize walls for surrounding as well as surrounding by whole hunter agents. When the surrounding is successfully performed, related hunter agents receive reward from the environment to carry out reinforcement learning. As for behavior of the prey agent, it behaves to run away from the nearest hunter agent for playing a fugitive role. In this study, we focus on the 4-agent pursuit problem to improve learning efficiency in multi-agent environment.

For simulation study, we adopt “soft-max” strategy for selecting the action of the hunter agents. The conditional probability based on Boltzmann distribution for action selection is as follows:

$$p(a|s) = \frac{\exp(w(s,a)/T_t)}{\sum_{a_1,a_2,...,a_N} \exp(w(s,a)/T_t)}$$

(1)

where $T_t$ is temperature at $t$-th iteration, $s$ is state vector, $a$ is the action of the agent, $b$ is the parameter for temperature cooling($0<b<1$), $w$ denotes evaluation value, and $N$ denotes the set of all alternative actions at the state $s$. Owing to this mechanism, the hunter agent act like random walk(exploring) in the early simulation trials and act definitely based on acquired evaluation values in the later simulation trials according to the lowered temperature value.

### III. A Hierarchical Reinforcement Learning Using a Modular Fuzzy Model Architecture

#### A. Basic Concept

There exist two problems to solve the pursuit problem efficiently. One is huge memory consumption for internal knowledge expression of the agents expressed as evaluation weights corresponding to the pair of state-and-action caused by the grid size of the environment and the number of hunter agents. In order to restrain the increase of required memory for the agents, modular structure is applied for expression of the agent knowledge base. The other is complex objective, i.e. surrounding the prey collaboratively. In general, it is effective for dealing with such the complex task to decompose into sub-tasks. Then we compose the task into hierarchical sub-tasks to fulfill reinforcement learning effectively. We propose a hierarchical modular reinforcement learning to solve the above described two problems in the multi-agent pursuit simulation.

#### B. Hierarchical Task Decomposition for Agent Learning

It is difficult to decide how many kinds of subtask should be decomposed into. In this study, we empirically decompose the surrounding task(capturing) into “decision of move position target” for surrounding according to current monitored state and “selection of appropriate action” to move to the target position of each agent. The latter task is native, isolated from the other hunter agents, and is not collaborative such as position control of the single agent. In other words, the task is decomposed into “surrounding” task synchronized with the other hunter agents and “exploring the environment” task. The upper task corresponds to collaborative surrounding strategy. Figure 3 shows the internal hierarchical structure of the hunter agent. The knowledge base of the agent is composed of the “Rules in Upper Layer” and the “Rules in Lower Layer” as shown in the figure. It is important to keep learning capability as well as task decomposition. According to the 2-layered decomposition, rules in the lower layer can be adopted corresponding to the agent behavior in every step as Markov Decision Process, as shown in Fig.4.

---

**Fig.2. Examples of Capturing Condition**

---

**Fig.3. Internal Hierarchical Structure of Hunter Agent**

---

**Fig.4. Conceptual Diagram of Hierarchical Task Decomposition**
C. A Modular Profit Sharing Learning for Upper Layer

In the upper layer, the target position of the agent is decided based on observed state such as the current position of the prey agent and the other hunter agents. The rules in the upper layer express goodness of the target position corresponding to the current state excluding actual actions. In order to construct the rules based on the current state combination, huge corresponding memory is needed. To avoid such requirement, we apply modular structure for the rule expression [16] in the upper layer as shown in Fig.5.

Original state space of each agent is expressed by covering by three subspaces of oneself-and-another pair as shown in Fig.6.

The weights of rules in the upper layer are updated by Profit Sharing learning algorithm, when capturing succeeds, as the following formulations:

\[
 u(e, g, h_{c,i}, h_{i}) = u(e, g, h_{c,i}, h_{i}) + k(e, g, h_{c,i}, h_{i})
\]

\[
 k(e, g, h_{c,i}, h_{i}) = \frac{1}{\rho} k(e, g, h_{c,i}, h_{i})
\]

\[
 (i = 0, 1, ..., m-1, e \neq c)
\]

D. Q-Learning for Lower Layer

In the lower layer, proper concrete action selection to reach the target position decided at the upper layer should be fulfilled through reinforcement learning process. It should be noted that states of the other hunter agents are unnecessary for the lower task. The input state of the rule consists of the target position and the current position of the hunter agent. At every step in learning trial, the learning of the lower layer is employed because we can interpret every agent movement as the movement to current position considered as the movement to the targeted position. In the lower layer, Q-Learning can be applied successfully because the process is typical Markov Decision Process. Q-Learning is realized as:

\[
 Q(s_{e,t}, a_{e,t}, c) = Q(s_{e,t}, a_{e,t}, c) + k \left( r + x \max \eta Q(s_{e,t}, \eta, c) - Q(s_{e,t}, a_{e,t}, c) \right)
\]

where \( Q \) is Q-value, \( s_{e,t} \) is the state vector of the agent \( e \) at \( t \)-th step, \( a_{e,t} \) is action of the agent \( e \) at \( t \)-th step, \( c \) denotes the state for updating, \( r \) denotes the reward, and \( k \), \( x \) are parameters. It should be noted that the current state of the agent moved from the other position always receive rewards considered as the targeted state.

IV. A MODULAR FUZZY MODEL

A. Model Structure

As a fuzzy model having high applicability, Single Input Rule Modules(SIRM)s[14][15] is proposed. The idea is to unify reasoning outputs from fuzzy rule modules comprised with single input formed fuzzy if-then rules. The number of rules can be drastically reduced as well as bringing us high maintainability in actual application. However, its disadvantage of low precision is inevitable in order to apply the method to huge multi-dimensional problems. We extend the SIRM method by relaxing the restriction of the input space,
We propose a “Modular Fuzzy Model”, for constructing the model of huge multi-dimensional space. Description of the model is as follows:

\[ \text{Rules} - 1 : \{ \text{if } P_i(x) \text{ is } A_j^i \text{ then } y_1 = f_j^i(P_i(x)) \}_{j=1}^{m_i} \]

\[ \vdots \]

\[ \text{Rules} - i : \{ \text{if } P_i(x) \text{ is } A_j^i \text{ then } y_i = f_j^i(P_i(x)) \}_{j=1}^{m_i} \]  

(5)

\[ \vdots \]

\[ \text{Rules} - n : \{ \text{if } P_n(x) \text{ is } A_j^n \text{ then } y_n = f_j^n(P_n(x)) \}_{j=1}^{m_n} \]

where “Rules-i” stands for the \(i\)-th fuzzy rule module, \(P_i(x)\) denotes predetermined projection of the input vector \(x\) in \(i\)-th module, \(y_i\) is the output variable, and \(n\) is the number of rule modules. The number of constituent rules in the \(i\)-th fuzzy rule module is \(m_i\). \(f\) is the function of consequent part of the rule like TSK-fuzzy model. \(A_j^i\) denotes the fuzzy sets defined in the projected space.

The membership degree of the antecedent part of \(j\)-th rule in “Rules-i” module is calculated as:

\[ h_j^i = A_j^i(P_i(x^0)) \]  

(6)

where \(h\) denotes the membership degree and \(x^0\) is an input vector. The output of fuzzy reasoning of each module is decided as the following equation.

\[ y_i^0 = \frac{\sum_{k=1}^{m_i} h_k^i \cdot f_j^i(P_k(x^0))}{\sum_{k=1}^{m_i} h_k^i} \]  

(7)

The final output of the “Modular Fuzzy Model” is formulated as:

\[ y^0 = \sum_{i=1}^{n} w_i \cdot y_i^0 \]  

(8)

where \(w_i\) denotes the parameter of importance of the \(i\)-th rule module. The parameter can also be formulated as the output of rule based system like modular network.

B. Application of Modular Fuzzy Model for Upper Layer

We tackle to the “Curse of Dimensionality” in the multi-agent pursuit problem using above proposed modular fuzzy model method. Our objective is to restrain memory consumption of rules in reinforcement learning keeping its performance. Instead of “Crisp Type” modular model described in section 3.C, we apply the modular fuzzy model to the upper layer model in the hierarchical reinforcement learning for pursuit problem. Fuzzy sets of the rules are defined as shown in Fig.7. The antecedent fuzzy sets are defined by Cartesian products of each fuzzy set on the state of the agent position.

\(u\) in (3) is calculated by the modular fuzzy model and is learned considering the membership degree of the rules by the profit sharing algorithm.
method is “crisp” modular model for upper layer. The number of rules in upper layer of each agent is \((25*25*25)*3= 46,875\). The last method is the fuzzy modular model for upper layer. For example, the 1st agent of the modular fuzzy model for upper layer is constructed as:

\[
\text{Rules } -1: \quad \{\text{if } [g,h_1,h_2] \text{ is } A_1^{j} \text{ then } y_1 = b_j^{129} \}
\]

\[
\text{Rules } -2: \quad \{\text{if } [g,h_1,h_1] \text{ is } A_1^{j} \text{ then } y_2 = b_j^{129} \}
\]

\[
\text{Rules } -3: \quad \{\text{if } [g,h_1,h_1] \text{ is } A_1^{j} \text{ then } y_3 = b_j^{129} \}
\]

where \(g\) is the position of the prey agent, \(h\) is the position of the hunter agent, and \(b\) is the parameter of consequent part of the fuzzy rule. The fuzzy set \(A\) is constructed by combining the fuzzy set of each agent position defined by partitioning the grid into 3x3 as shown in Fig.7. The number of rules in upper layer is much smaller than the others, i.e. \((9*9*9)*3=2,187\).

VI. CONCLUSION

In this paper, we focus on the pursuit problem and propose a hierarchical modular reinforcement learning that Profit Sharing learning algorithm is combined with Q Learning reinforcement learning algorithm hierarchically in multi-agent environment. As the model structure for such the huge problem, we propose a modular fuzzy model extending SIRMs architecture. Through numerical experiments, we show the effectiveness of the proposed algorithm compared with the conventional algorithms.

We perform the simulation 20 times for each method. The results are shown in Fig.9. The depicted data is averaged value of 20 series after averaging each sequential 100 trials. The results by “crisp” modular model(CrispMod) show the best performance compared with the other methods. The modular fuzzy model(FuzzyMod) is not inferior to the complete expression model(NonMod). The number of the rules in the modular fuzzy model seems inadequate because of the deterioration of the performance in later trials. However the learning speed is as good as the crisp modular model from the results of early trial stage. From these results, it can be said that the modular fuzzy model structure is promising for high dimensional problem, depending upon its usage. The simple Q-Learning agent (NonH-Q) is not so bad in the small 5x5 grid world. The strategy only to approach the prey agent acquired by the simple non-hierarchical Q-Learning might be reasonable in such the small world.

REFERENCES

[1] R. S. Sutton and A. G. Barto, “Reinforcement Learning”, MIT Press, 1998.
[2] C. J. H. Watkins and P. Dayan, “Technical Note: Q-Learning”, Machine Learning, Vol.8, pp.58-68, 1992.
[3] J. J. Grefenstette, “Credit Assignment in Rule Discovery Systems Based on Genetic Algorithms, Machine Learning, Vol.3, pp.225-245, 1988.
[4] K. Miyazaki, H. Kimura, and S. Kobayashi, “Theory and Application of Reinforcement Learning Based on Profit Sharing”, J. of JSII, Vol.14, No.5, pp.800-807, 1999.
[5] K. Miyazaki, S. Arai, and S. Kobayashi, “A Theory of Profit Sharing in Multi-agent Reinforcement Learning”, J. of JSII, Vol.14, No.6, pp.1156-1164, 1999.
[6] M. Benda, V. Jagannathan, and R. Dodhiwalla, “On Optimal Cooperation of Knowledge Sources”, Technical Report, BCS-G2010-28, Boeing AI Center, 1985.
[7] A. Itô and M. Kanabuchi, “Speeding up Multi-Agent Reinforcement Learning by Coarse-Graining of Perception –Hunter Game as an Example–”, Trans. of IEICE, Vol.84-D-I, No.3, pp.285-293, 2001.
[8] Y. Takahashi and M. Asada, “Behavior Acquisition by Multi-Layered Reinforcement Learning”, Proc. of the 1999 IEEE Int. Conf. on Syst., Man, and Cybern., pp.716-721, 1999.
[9] J. Morimoto and K. Doya, “Acquisition of Stand-up Behavior by a Real Robot using Hierarchical Reinforcement Learning”, Proc. of International Conference on Machine Learning, pp. 623-630, 2000.

[10] N. Ono and K. Fukumoto, “Multi-agent Reinforcement Learning: A Modular Approach”, Proc. 2nd Int. Conf. on Multi-agent Systems, pp.252-258, AAAI Press, 1996.

[11] K. Fujita and H. Matsuo, “Multi-agent Reinforcement Learning with the Partly High-Dimensional State Space, Trans. of IEICE, Vol.J88-D-1, No.4, pp.864-872, 2005.

[12] H. Murao and S. Kitamura, “Q-Learning with Adaptive State Segmentation(QLASS)”, Proc. IEEE Int. Symp. on Computational Intelligence in Robotics and Automation, pp.179-184, 1997.

[13] T. Hamagami, S. Koakutsu, and H. Hirata, “An Adjustment Method of the Number of States on Q-Learning Segmenting State Space Adaptively”, Trans. of IEICE, Vol. J86-D1, No.7, pp.490-499, 2003.

[14] H. Seki, H. Ishii, and M. Mizumoto: "On the Generalization of Single Input Rule Modules Connected Type Fuzzy Reasoning Method", Proc. of the SCIS&ISIS2006, pp.30-34, 2006.

[15] N. Yubazaki, J. Yi, M. Otani and K. Hirota, "SIRM's Dynamically Connected Fuzzy Inference Model and Its Applications, Proc. IFS’97, vol.3, pp.410-415, 1997.

[16] Y. Takahashi, T. Watanabe: "Learning of Agent Behavior Based on Hierarchical Modular Reinforcement Learning", Proc. of the SCIS&ISIS2006, pp.90-94, 2006.