International Journal of Fuzzy Logic and Intelligent Systems, vol. 5, no. 2, June 2005 pp. 151-156

Learning of Cooperative Behavior between Robots in Distributed Autonomous Robotic System

Chel-Min Hwang and Kwee-Bo Sim

School of Electrical and Electronic Engineering, Chung -Ang University

Abstract

This paper proposes a Distributed Autonomous Robotic System(DARS) based on an Artificial Immune System(AIS) and a Classifier System(CS). The behaviors of robots in the system are divided into global behaviors and local behaviors. The global behaviors are actions to search tasks in given environment. These actions are composed of two types: aggregation and dispersion. AIS decides one among these two actions, which robot should select and act on in the global. The local behaviors are actions to execute searched tasks. The robots learn the cooperative actions in these behaviors by the CS in the local one. The proposed system will be more adaptive than the existing system at the viewpoint that the robots learn and adapt the changing of tasks.

Key words : Distributed Autonomous Robotic System, Artificial Immune System, Classifier System

1. Introduction

In the past, we used centralized control method to manage a robot control system. This method was fast and accurate because this simultaneously manages the system in the center. However, the systems are getting bigger and more complex as technologies are rapidly developed and enlarged and robustness and flexibility in the control were required. The many algorithms were proposed to satisfy those necessities and distributed autonomous system[1-2], which was based on a society of human, a group of insect, a biological immune system, and so on was one of them. The distributed autonomous robotic system(DARS) performs cooperative behavior of multi-agent robots by using those algorithms.

In this paper, we propose an adaptive system for unknown environment by adding the learning algorithm in DARS. We use an artificial immune network based on an artificial immune system[3-5] and a classifier system[6-13] to compose the proposed system. The distributed autonomous robotic system, which is composed by artificial immune network, is able to do adaptive behaviors by selecting proper action for dynamic environment[14-15]. Also, the classifier system can receive bigger rewards from unknown environment by learning.

The artificial immune network is equations, which are modeling reactions between antigen and antibody in the immune system. In the relation of between the antigen and the antibody, B-cells that have been stimulated by binding to their specific antigen take the first steps towards cell division. They express new receptors that allow them to respond to cytokines from other cells, which is signal proliferation. The B-cells may also start to secrete cytokines themselves. They will usually go through a number of cycles of division, before differentiating into mature cells, again under the influence of cytokines. For example, proliferating B cells eventually mature into antibody-producing plasma cells. Even when the infection has been overcome, some of the newly produced lymphocytes remain, available for re-stimulation if the antigen is encountered once more. These cells are called memory cells, since they retain the immunological memory of particular antigen. It is memory cells that confer the lasting immunity to a particular pathogen[3-5]. We can compose Distributed Autonomous Robotic System by setting the antigen and antibody in those equations, which express previous processes

The classifier system that is one of the machine learning finds useful things in many rules, which mean the action for the environments. The system uses 'bucket brigade algorithm' to allocate strength of rules and searches another things from allocated them[6-8]. We use XCS[10-13] among many classifier systems in this paper. The XCS, which is proposed to solve problems of previous classifier systems, is similar form with Zeroth level Classifier System (ZCS)[6-9] but more advanced than that. The previous CS has a disadvantage, which is that Genetic Algorithm (GA) disrupts the useful rules having low strength easily because the systems use strength as fitness when they select the rules. The XCS uses 'predict', 'error', and 'fitness' instead of 'strength' and progresses GA not in the 'population' but in the 'Action Set'. Additionally, the number of using rules is not fixed and the rules are added and deleted by need of the system.

Manuscript received Mar. 13, 2005; revised June. 1, 2005
This research was supported by the Brain Neuroinformatics Research Program sponsored by Korea Ministry of Science and Technology.

The system divides the actions into a global behavior and a local behavior according to the action is either based on the artificial immune network, or based on the CS to adapt changing environment. The global behavior based on the artificial immune network selects action of robot to make up the environment, which is that the robots search tasks and execute that fastly. The local behavior based on the CS detects type of task and decides the action of the robot according to type of that. Then, CS learns roles of robot according to efficiency of executing tasks. This system gets better conclusion when it detects unknown tasks because of the referred process.

2. Artificial Immune Network(AIN)

2.1. Outline of AIN

An immune network is relation between antigen and antibody. Its exact operation is not completely proved yet. However, there are some useful hypotheses about that. In this paper, we use artificial immune network which is proposed by 'Jerne' who is immunologist[3]. The artificial immune network hypothesis proposed by 'Jerne' is proved that it is not true, but it has sufficiently worth because that similarly describes the function of biological immune system in result. Next equations express artificial immune network, which is applied to Distributed Autonomous Robotic System in this paper[14].

$$S_i(t+1) = S_i(t) + (\alpha \frac{\sum_{j=1}^{N} m_{ij} s_j(t)}{N} + \beta g_i(t) - c_i(t) - k_i) s_i(t) \qquad (1)$$

$$s_i(t) = \frac{1}{1 + \exp(0.5 - S_i(t))} \qquad (2)$$

$$c_i(t) = \eta(1 - g_i(t)) S_i(t) \qquad (3)$$

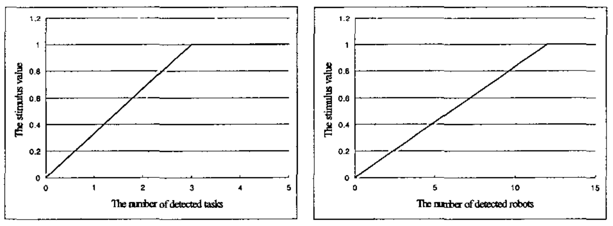
where $i, j = 0, \ldots, N\text{-}1$, N is a number of antibody types, $S_i(t)$ is stimulus value of antibody i, $s_i(t)$ is concentration of antibody i, $s_j(t)$ is not concentration of self-antibody but that of other robot's antibody obtained by communication, $c_i(t)$ is concentration of T-cell which control concentration of antibody, m_{ij} is mutual stimulus coefficient of antibody i and j, $g_i(t)$ is affinity of antibody i and antigen, α, β, and η are constants.

In the equation (3), when the stimulus value of antigen is big and the stimulus value of antibody is small, the concentration of T-cell is small. Therefore, in this case $c_i(t)$ take a role of helper T-cell that stimulates B-cell. On the contrary, the stimulus value of antigen is small and the stimulus value of antibody is big, the $c_i(t)$ is big. So, it takes part in suppressor T-cell. In biological immune system the helper T-cell activate B-cell when the antigen invade it, and the suppressor T-cell prevent the activation of B-cell when the antigen was eliminated. By adding T-cell modeling, performance of system (making group behavior) is improved[3][14].

2.2 The Operation of Immune System

The action in the global behavior is selected by artificial immune network in this paper. The global behavior of each robot is process of searching tasks and gathering robot to execute that. Aggregation and dispersion are the actions of the global behavior. Each robot decides which action the global behavior after detecting near environment and judging condition. That there are too many robots in the narrow space is like reduction of searching space. So, the system makes easy finding tasks by dispersing robots when many robots assemble at around tasks. Also, the system aggregates robots to execute task when the robot detects a task. Then, deciding the action of a robot in global behavior is very important. So, we apply artificial immune network equation to the system as the antigens are the tasks and the robots and the antibody are the actions of global behavior.

In order to apply the immune network equation, the stimulus value of antigen should be set. This value depends on a number of tasks and robots as it shows in the Fig. 2. The value in the Fig. 2(a) is established by using that how long tasks last in the detect range. The value is 1 when tasks are always sensed during 10 tests, the value is 0 when tasks are never sensed during tests and the values between these two are decided by linier line. This is in order to accomplish tasks by using more robots as the robot judges that the task is not accomplished when the many tasks are detected in the range. In this case, robots working on tasks are constantly stimulated but this is ignored because tasks robots are working on disappear after they are completed. The value in the Fig. 2(b) is established by using that how many robots exist in the detect range. The value is 1 when a number of robots are more than 4, the value is 0 when there is not any robot and the values between these two are also decided by linier line. The robot judges that the number of working robots is too many when more than 5 robots work together in detect range and selects dispersion to reduce the number of robots. The stimulus values between antibodies are showed in the table 1 and this is based on a mutual stimulation between the same antibody, and a little bit of suppression between the different antibodies.



(a) (b)

Fig. 2. The stimulus function of antigen that stimulates antibodies, which are related to (a) Aggregation and (b) Dispersion.

Table 1. Mutual stimulus coefficients

m_{ij}	Aggregation	Dispersion
Aggregation	1	−0.1
Dispersion	-0.1	1

3. Classifier System(CS)

3.1 Outline of CS

The CS is one of the genetic algorithm based machine learning and an adaptive system. This system proposed by 'Holland' learns rules which is called as the classifier. The classifier is composed of 'condition', 'action', and 'strength'. The 'condition' is represented by a ternary bit string that is set of {0, 1, #}, and the bit string of 'action' is set of {0, 1}. Also, the strength means a usefulness of rule. The leaning system is divided into three subsystems, which are performance system, learning system, and rule discovery system. The performance of the system is composed of three parts, which are encoder, rule base, and decoder. The encoder generates a message, which have equal string length with 'condition'. The rule base has a group of classifiers, which is called rule set, and generates a message from comparing the message of encoder with the rule set. The decoder expresses the message of rule base to environment. The learning is accomplished by a repetition of this cycle. If the rule base can't generate a message, it adds a new classifier at the rule set. The 'condition' of added classifier is generated from message of encoder but some of string is altered into '#'. The 'action' is generated from random. Then another classifier is deleted to maintain size of rule set. The learning system assigns reward by 'bucket brigade algorithm'. Therefore, we may regard the 'strength' as the prediction of reward when the system selects that classifier. The rule search system is based on the genetic algorithm. The genetic algorithm searches new rules based on fitness, which is previously defined. The fitness of classical CS directly uses the strength of classifier. The algorithm generates new classifier from selected classifier and adds at the rule set. Next, it deletes as many classifier as they has been added. Then, the selection is based on strength, too[6-9].

3.2 Outline of XCS

The XCS, which is proposed by 'Wilson', is a kind of CS. The classical CS has a disadvantage that is the CS disrupts the useful rules, which have low strength, because the system uses the strength as the fitness. The XCS uses prediction, error, and fitness instead of strength to solve this problem. Also, genetic algorithm applied to action set, not rule set, which is sub set of rule set [12-13]. A detail would be explained later. Additionally, the number of classifier is properly kept up by rule addition and deletion.

Fig. 2 shows schematic illustration of XCS. The system recognizes change of environment and transforms them into string message, which is used in oneself. This message compares with the 'condition' of classifiers and fired classifiers - both are equal except '#' bits - become match set. The classifiers in match set are divided as to 'action' and divided classifiers compose a prediction array from prediction of classifiers. The prediction array becomes basis of selecting action. The selection is either probable selection or maximum selection. The classifiers, which have selected action in match set, compose action set. The action in action set is expressed at the environment by decoder. The reward from environment and maximum prediction in prediction array are used at update of parameters, which are prediction, error, and fitness, of previous action set. Finally, new rules are added and useless rules are deleted by genetic algorithm in previous action set.

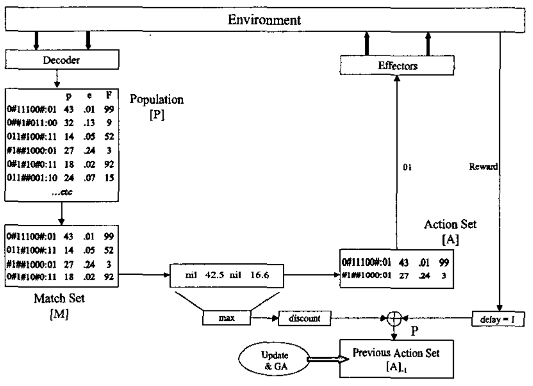


Fig. 2. The schematic illustration of XCS

The several parameter update equations is as follows:

$$p \leftarrow p + \beta(P - p) \tag{4}$$

$$\varepsilon \leftarrow \beta(|P - p| - \varepsilon) \tag{5}$$

$$k \leftarrow \exp[(\ln\alpha)(\varepsilon - \varepsilon_0)/\varepsilon_0] \tag{6}$$

$$F \leftarrow F + \beta(k' - F) \tag{7}$$

In above equations, the learning rate β has real number between zero and one. The k means accuracy for prediction of classifier and then, and the ε_0 is an allowable error. If error is lower than the allowable error, the prediction is regarded as correct. The α, which is discount factor of accuracy when error is bigger than allowable error, has real number between zero and one, too. This concept of updates is similar with 'Q-learning' than 'bucket brigade algorithm' [10-13].

The previous CS does not admit overlapping among classifiers and doesn't add classifiers, which are newly generated, if rule set has them. However, XCS admits overlapping and redefines overlapped classifier as macro-classifier. Each macro-classifier has numerosity parameter, which means the number of overlapped classifier. If numerosity of classifier is zero then, the classifier is deleted from rule set. If rule set has same, its numerosity increases when the classifier is added. Also, the initial value of numerosity is one. The use of macro-classifier reduces using classifier in rule set and improves the accuracy of reminded classifiers. We can regard the distribution of macro-classifier as the complexity of solution in search space.

The system reduces the number of macro-classifier by maximizing the classifiers in the rule set. If a classifier represents more detail, it may predict more accurate prediction. Hence, the system satisfies a precondition, which is that the accuracy of rule doesn't decrease, to generalize classifier. The system allows next procedure to satisfy the precondition. The system selects a classifier, which has high accuracy, and copies it. Replace one bit, which is '0' or '1', in 'condition' of copied classifier as '#'. The new classifier has higher selection probability because it is more generalized than original. If the generalization of new classifier is incorrect, the accuracy is lower and lower in progress of time. If it is correct, the accuracy will be higher than original. Consequently, the classifier, which is more generalized and more accurate, will survive.

3.3 Application of XCS

The XCS decides the local behavior in the system. The local behavior represents the role of robots that execute tasks. The type of tasks is four and action of robot to execute task are also four. The four robots handle one task at the same time to execute it. The rule of classifier is expressed binary string of which length is six bits. First four bits are 'condition' and second two bits are 'action'. The 'condition' is composed of the type of task and the sequence which robot discovers. The 'action' is composed of role of robot. The paper of 'Wilson' uses the most parameters of the system. The size of rule set is 100 and generation probability of '#' is set 0.3.

4. Simulation

4.1 Simulation environment

We propose that the robots detect tasks from environment and execute it as shown Fig. 3. A task is executed when four robots simultaneously accomplish it. Also, there is suitable combination for role of robots to get best performance. Each robot selects one role among four when it executes a task. The simulation is accomplished three ways. The positions of twenty tasks are randomly decided. The mass of a task is ten and the executing size for one time is one or less. If the execution of a task is completed, the task is regenerated with initializing the mass of it. Therefore, the number of tasks is fixed. A robot selects a role to execute the task from the type and position of oneself for execution. Then, the efficiency between the accomplished role and necessary role is expressed at table 2. The evaluation for performance is based on how fast the robots execute the task. In the first simulation, we use two types task and the number of each is ten. In the second, the types of tasks are periodically changed. In the last, we expand the number of task to four.

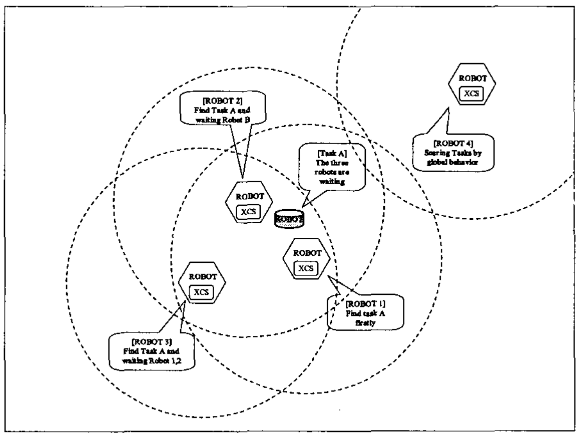

[ROBOT 2] Find Task A and waiting Robot B
[Task A] The three robots are waiting
[ROBOT 4] Searing Tasks by global behavior
ROBOT
XCS
ROBOT
XCS
ROBOT
ROBOT
XCS
ROBOT
XCS
[ROBOT 1] Find task A firstly
[ROBOT 3] Find Task A and waiting Robot 1,2

Fig. 3. Simulation environment.

Table 2. The efficiency between accomplished role and necessary role.

	R0	R1	R2	R3
R0	1	0.5	0	0.5
R1	0.5	1	0.5	0
R2	0	0.5	1	0.5
R3	0.5	0	0.5	1

4.2 The simulation results

We measure the executed mass of a task for a time in the simulation environment as a performance of the system. The performance of execution is measured one or less. We display the performance of the system as a result of simulation at Fig. 4-6. The value of the performance is calculated from average value for one thousand times. The crosswise-axis means a flow of time and the vertically-axis means average performance of the system in the figures. Fig. 4 shows the result of first simulation. The performance of system is improved from 0.5 to 0.8 and slightly oscillates after convesing at 0.8. This is regarded as improvement of performance for the detected tasks. Also the oscillation is occurred by exploration of learning system. Fig. 5 shows the result of second simulation. The type of generated tasks are changed every a thousand times and whole period is two thousands. We can observe that the landscape falls down when the system starts and changes the tasks firstly and slightly increases to convergence. However, the second and third don't fall down but change to oscillation because the system has the classifiers, which are learned previously. Fig. 6 is the result of last simulation. It shows the similar result with Fig. 4. However, we can recognize that the converged value is lower and time is later than it.

This is influence of increased tasks. The increased tasks influence the number of macro-classifier. Hence, the selection probability for suitable role is lower. Additionally, the converged value doesn't reach one because the rewards were not evaluated independently and each robot learns by both exploitation and exploration.

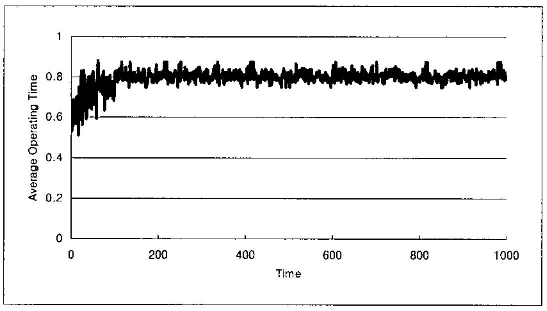


Fig. 4. Average speed of operation in two tasks.

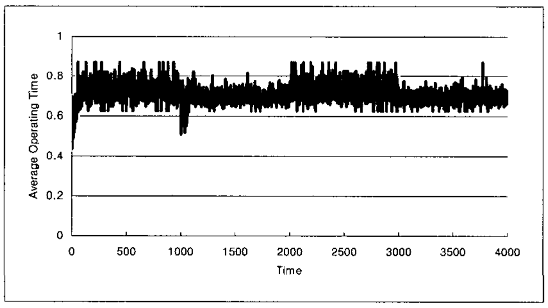


Fig. 5. Average speed of operation in periodic change of type of two tasks.

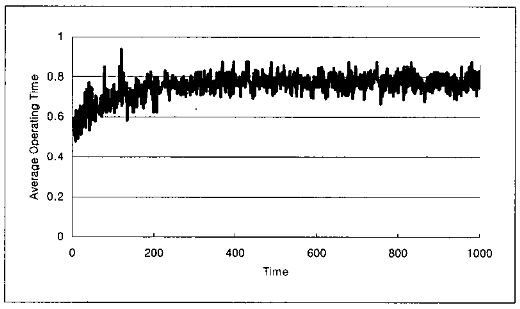


Fig. 6. Average speed of operation in four tasks.

5. Conclusion

Recently, an interest for autonomous robotic system is increased rapidly. The DARS is a specific system, which wants to get better performance through the cooperative behaviors among the robots, of this system. The DARS will be needed more in the future, when the autonomous robots become a popular system in the human life to accomplish the works, which a robot can't do alone. However, fixed programs of the robots can't prepare all situations in such environment. Therefore, the robots need a learning system. In this paper, we propose the learning methods in DARS and evaluate the proposed system in the simulation. In the proposed system, the robots do not communicate the information of environment between themselves. However, they decide a proper action from its environment. The performance of the system is not good but shows that the system copes with unknown environment and changes. Consequently, embodying adaptive DARS, which is based on AIN and XCS, is possible. Nevertheless, the study for how we improve the performance of the system must be continued.

References

[1] H. Asama, "Trends of Distributed Autonomous Robotic Systems," *Distributed Autonomous Robotic Systems*, vol. 1, pp. 3-8, 1994.

[2] J. Kennedy, R. C. Eberhart, Y. Shi, *Swarm Intelligence*, SanFrancisco : Morgan Kaufman Publishers, 2001

[3] N. K. Jerne, "Idiotopic Network and Other Preconceived Idias," *Immunological Rev.*, vol. 79, pp. 5-24, 1984.

[4] D. Dasgupta, N. Attoh-Okine, "Immunity-Based Systems: A Survey," Systems, Man, and Cybernetics, Computational Cybernetics and Simulation., *1997 IEEE International Conference*, vol. 1, pp. 369-374, 1997.10.

[5] H. Meshref, H. VanLandingham, "Artificial Immune Systems; Application to Autonomous Agents," Systems, Man, and Cybernetics, *2000 IEEE International Conference on*, vol. 1, pp. 61-66, 2000. 1.

[6] G. W. Flake, *The Computational Beauty of Nature: Computer Explorations of Fractals, Chaos, Complex Systems, and Adaptation*, Cambridge, Mass: MIT Press, 1998.

[7] T. Kovacs, "What Should a Classifier Learning," *Evolutionary Computation, Proceedings of the 2001 Congress*, Vol. 2, pp. 775-783, 2001.

[8] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison Wesley, 1989.

[9] S. W. Wilson, "ZCS: A Zeroth Level Classifier System," Evolutionary Computation, Vol. 2, No. 1, pp. 1-18, 1994

[10] S. W. Wilson, "Classifier Fitness Based on Accuracy," Evolutionary Computation, Vol. 3, No. 2, 1995

[11] M. V. Butz, S. W. Wilson, "An Algorithmic Description of XCS," Lecture Notes in Computer Science, Vol. 1996, pp. 253-270, 2001

[12] S. W. Wilson, "Generalization in the XCS Classifier System," *Genetic Programming Proceedings of the Third Annual Conference*, pp. 665-674, 1998.

[13] T. Kovacs, "Evolving Optimal Populations with XCS Classifier Systems," *MSc. Dissertation, Univ. of Birmingham, UK.*, 1996.

[14] D.W. Lee, K.B. Sim, "Behavior Learning and Evolution of Collective Autonomous Mobile Robots using Distributed Genetic Algorithms," *2nd Asian Control conference*, vol. 2, pp. 675-678, 1997. 7.

[15] M. Kaiser, V. Klingspor, J. R. Millan, M. Accame, F. Wallner, R. Dillmann, " Using Machine Learning Techniques in Real-World Mobile Robots," *Expert, IEEE [see also IEEE Intelligent Systems]*, vol. 10, Issue 2, pp.37-45, 1995.

Chul-Min Hwang
Chul-Min Hwang received the B.S. degrees in the School of Electrical and Electronic Engineering from Chung-Ang University, Seoul, Korea, and is currently working in the master's course. His research interests are Artificial Life, Evolutionary Computation, Distributed Autonomous Robotic Systems, Machine Learning, and Artificial Intelligence.

Kwee-Bo Sim
Kwee-Bo Sim received his B.S. and M.S. degrees in the Department of Electronic Engineering from Chung-Ang University in 1984 and 1986 respectively, and Ph.D. degree in the Department of Electrical Engineering from The University of Tokyo, Japan, in 1990. Since 1991, he has been a faculty member of the School of Electrical and Electronic Engineering at Chung-Ang University, where he is currently a Professor. His areas of interest include artificial life, neuro-fuzzy and soft computing, evolutionary computation, learning and adaptation algorithms, autonomous decentralized systems, intelligent control and robot systems, artificial immune systems, evolvable hardware, and artificial brain etc. He is a member of IEEE, SICE, RSJ, KITE, KIEE, ICASE, and KFIS.

Phone	: +82-2-820-5319
Fax	: +82-2-817-0553
E-mail	: kbsim@cau.ac.kr