Adaptive Distributed Autonomous Robotic System based on Artificial Immune Network and Classifier System

Chul-Min hwang and Kwee-Bo Sim

School of Electrical and Electronic Engineering, Chung-Ang University, Seoul, Korea
(Tel : +82-2-820-5319; E-mail: kbsim@cau.ac.kr)

Abstract: This paper proposes a Distributed Autonomous Robotic System (DARS) based on an Artificial Immune Network (AIN) 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 environment. These actions are composed of two types: aggregation and dispersion. AIN decides one between 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. The relation between global and local increases the performance of system. Also, the proposed system is more adaptive than the existing system at the viewpoint that the robots learn and adapt the changing of tasks.

Keywords: Distributed Autonomous Robotic System, Artificial Immune Network, Classifier System

1. INTRODUCTION

Centralized control system is used to control systems in the past. This control system has an advantage that is fast and accurate control is possible because it manage all parts as necessary procedure. However, the systems have been enlarge and complicated by rapid growth of technologies in the latest. Consequently, flexibility and robustness become more important than before. So, many algorithms are proposed to satisfy it. A distributed autonomous system, which we use in here, is one of them. The distributed autonomous systems are modeled on human society, a group of insect, or biological immune system. The DARS is cooperative behavior of multi-agent robots, which are based on distributed autonomous system [1].

In this paper, we propose an adaptive system for varying tasks by appending learning algorithm to DARS. The proposed system is composed of artificial immune network (AIN), which is based on AIS, and CS. The DARS, which is based on AIN, can do adaptive behaviors for changing environment. Also, CS is able to learn for unknown environment and select proper actions that obtain more reward.

AIN is expressed an equation, which mean a reaction of between antigen and antibody. In this reaction, a B-cell becomes activation from stimulation of a specific-antigen. Then, a B-cell is stimulated by only one species of antigen. The activated B-cell generates memory cell or transform oneself into plasma cell. A helper T-cell, which promotes this sequence, of which operation begins by promotion of macrophages binding an antigen. The plasma cell secretes antibody, which is able to recognize the invasive antigen. A suppressor T-cell disturbs the operation of plasma cell when the antibodies can't bind antigens any more. AIN expresses relation between B-cell and T-cell as equations [2]. We can apply this equation to DARS by defining suitable antigens and antibodies [3-5].

The CS, which is one of Machine Learning, searches useful rules in every rule that is action for detected environment. It uses 'bucket brigade algorithm' to update a strength of rules and updated strengths become the basis of rule search. We use XCS in this study. The XCS is not only similar with Zeroth level Classifier System, which is recently proposed system to solve problems of classical CS and its framework but also advanced model. When the system searches new rules by the genetic algorithm, the previous CS has a disadvantage, which is that useful rules are easily destroyed. The rule search algorithm is based on strength of itself but low strength does not mean useless rule because strength often implies a prediction value of reward. XCS use prediction, error, and fitness parameters instead of strength and the genetic algorithm is operated in not the rule set but the action set. Additionally, the system allows overlapping of rules in the population, therefore we can think that the number of using rule is not defined but varies.

We divide the operation of system into the global behavior, which is the searching system by the AIN and the local behavior, which is the learning system by the XCS. In the global behavior, the AIN selects the strategy that helps the robots looks for tasks and make environment of execution. In the local behavior, the XCS decides robots' action form detected environments. Then, the system learns the robot's reaction for the environments and improves the performance of execution. These structures take out better results for unknown tasks.

This research was supported by the Braintech 21 from Ministry of Science and Technology of Korea.

2. ARTIFICIAL IMMUNE NETWORK

2.1 The operation of immune system

Each lymphocyte (whether B-cell or T-cell) is genetically programmed to be capable of recognizing essentially only one particular antigen. The immune system as a whole can specifically recognize many thousands of antigens, so the lymphocyte recognizing any particular antigen must represent only a minute proportion of total. How then is an adequate immune response to an infectious agent generated? The answer is that when an antigen binds to the few cells that can recognize it, they are induced to proliferate rapidly. Within a few days, there are a sufficient number to mount an adequate immune response. In other words, the antigen selects for and generates the specific clones of its own antigen-binding cells, a process called clonal selection. This operates for both B-cells and T-cells [3-5].

Lymphocytes 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 cytokins from other cells, which signal proliferation. The lymphocyte may also start to secrete cytokins themselves. They will usually go through a number of cycles of division, before differentiating into mature cells, again under the influence of cytokins. 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].

Tolerance to self-antigens is established during ontogeny. However, during the neonatal period the unique binding regions of antigen-specific receptors on B-cell (antibody) and T-cells are present at levels that are too low to generate tolerance. This antigen-specific site is called idiotope. Thus antibodies (B-cells) are stimulated not only antigens but also other antibodies (B-cells). Jerne who is an immunologist proposed idiotopic network hypothesis (immune network hypothesis) based on mutually stimulus and suppression between antibodies. According to this hypothesis, antibodies interact with each other through idiotope and paratope. Such a network relationship plays a important role that needed concentration of antibody keeps up in the immune system. In this way, immune system is parallel distributed system that operates not unit level but system level [3-5].

2.3 The Artificial Immune Network

The operation of AIN is not proved accurately yet but there are some hypothesises. In this paper, we use the equation of the AIN, which is proposed at first by 'Jerne', immunologist, and added T-cell model. The AIN model of 'Jerne' is proved wrong model. Nevertheless, the network has sufficient worth because it satisfies the operation of immune system. Next equations are that we used in the DARS [3-5].

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

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

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

$S_i(t)$ $s_j(t)$ $c_i(t)$ m_{ij} $g_i(t)$

where i, j = 0,…,N-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,η are constants.

In 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.

2.2 The application of artificial immune network

In this paper, we decide the global behavior of the robots by the AIN. The global behavior of each robot searches tasks and recruits needed robots. We present two actions, aggregation and dispersion, to accomplish such purpose. Each robot detect sits environment and decide the global behavior. If distances between robots are short when there isn't task in the environment, the robots can't search tasks properly. So, the robots select dispersion as action and increase the efficiency of search. In contrast, if there aren't enough robots to execute a task when they accomplish it, they can't execute. Therefore, the robots, of which position is near with the task, select aggregation. The problem that the robots select one of them is very important. So, we regard the number of tasks and robot in detected environment as the antigen and the actions as antibody to apply to the AIN.

We must set up stimulus values of the antigen before application of the AIN. The stimulus values of antigen are based on the number of tasks and robots. Figs. 1,2 shows the graph of the setting of antigen. The Fig. 1 is the stimulus value of an antigen, which stimulates the aggregation, is set from how many tasks are detected in the environment. If the number of tasks is equal or more than three, the stimulus value is 1 and if there is not any task, the value is 0. Otherwise, linearly set from 0 to 1.

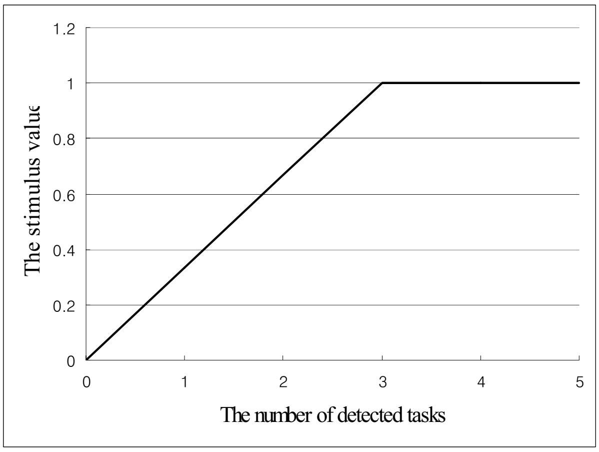

Fig. 1 The stimulus function of antigen that stimulates antibodies, which are related to aggregation.

The Fig. 2 is the stimulus value of an antigen, which stimulates the dispersion, is set from how many robots are detected in the environment. If the number of tasks is equal or more than twelve, the stimulus value is 1 and if there isn't any task, the value is 0. Otherwise, linearly set from 0 to 1. The robots select high between two stimulus values. The mutual stimulus coefficient of antibody is represented at table 1. The same antibodies stimulus mutually and different antibodies suppress each other. The constants of the equations are set like α = 0.001, β = 0.8, η = 0.05, and k_i = 0.1.

Table 1. Mutual stimulus coefficients

	Aggregation	Dispersion
Aggregation	1	−0.1
Dispersion	-0.1	1

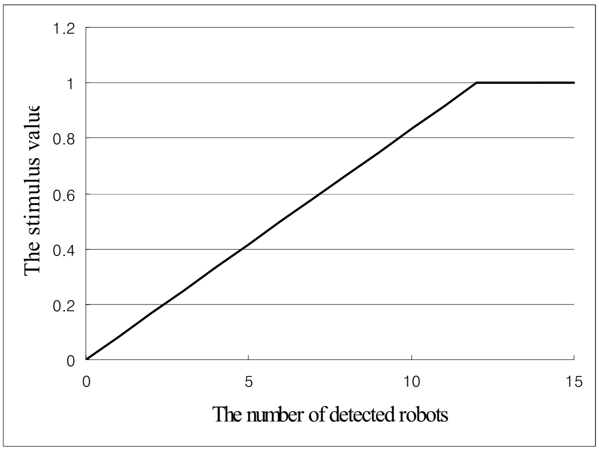

Fig. 2 The stimulus function of antigen that stimulates antibodies, which are related to dispersion.

3. CLASSIFIER SYSTEM

3.1 Preview

The CS is one of the genetic based machine learning and an adaptive system. This system, which is proposed by 'Holland', is learning for the rules, which are called 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 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 repetition of this cycle. If the rule base can't generate message, it adds 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 other 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 classifiers as many as added thing. Then, the selection is based on strength, too [6-9].

3.2 XCS

The XCS, which is proposed by 'Wilson', is one 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 [7-9]. A detail would be explained in next. 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.

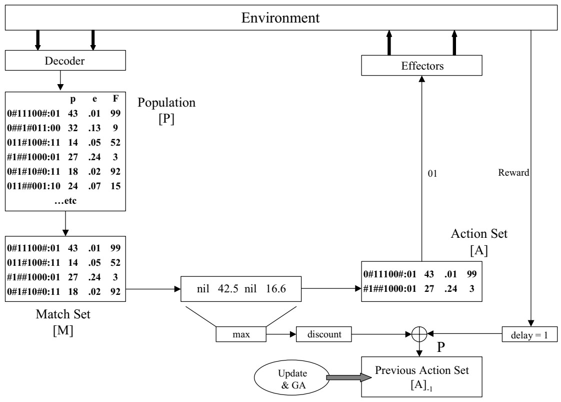

Fig. 2. The schematic illustration of XCS.

Below are the parameter update equations.

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

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

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

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

In these equations, learning rate, β , has real number between zero and one. k means accuracy for prediction of classifier and then, ε_0 is an allowable error. if error is lower than the allowable error, the prediction is regarded as correct. 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'[6-9].

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 macroclassifier. Each macroclassifier has numerosity parameter, which means the number of overlapped classifier. if numerosity of classifier is zero then, the classifire 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 macroclassifier reduces using classifier in rule set and improves the accuracy of reminded classifiers. We can regard the distribution of macroclassifier as the complexity of solution in search space.

The system reduces the number of macroclassifier 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 at simulation

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 almost parameters of the XCS are used by the paper of 'Wilson'. The size of rule set is 100 and generation probability of '#' is set 0.3.

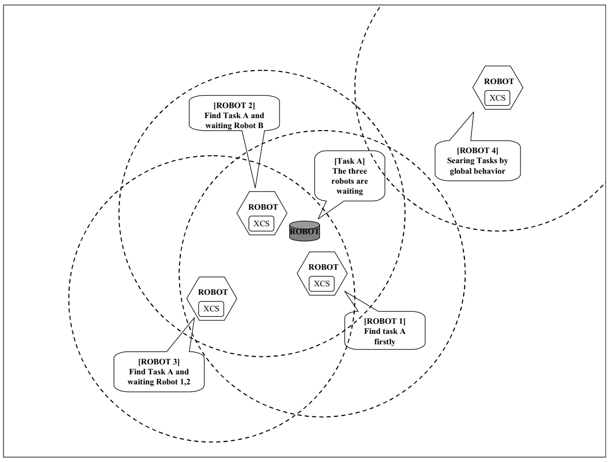

Fig. 3 The simulation environment.

4. SIMULATION

4.1 Simulation environment

We propose that the robots detect tasks from environment and execute it. 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 for 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.

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 0.6 and slightly oscillating. This is regarded as improvement of performance for the detected tasks. Also the oscillation is occurred by exploration of learning system.

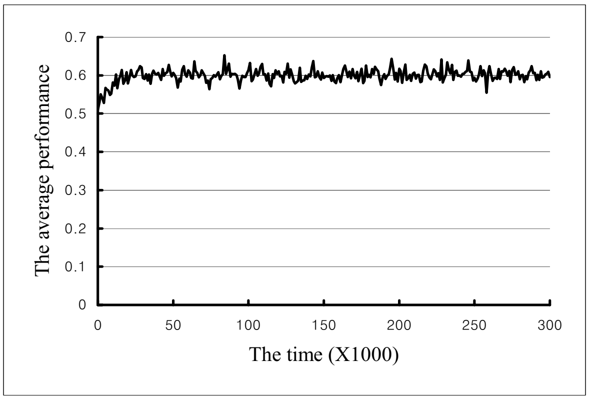

Fig. 4. Average speed of operation in two tasks.

Fig. 5 shows the result of second simulation. We can observe the periodic landscape every one hundred times. The performance rapidly falls down and slightly grows up. This phenomenon is the result for that the system adapts the changing environment. The performance falls down when the type of the tasks is changed and grows up by learning.

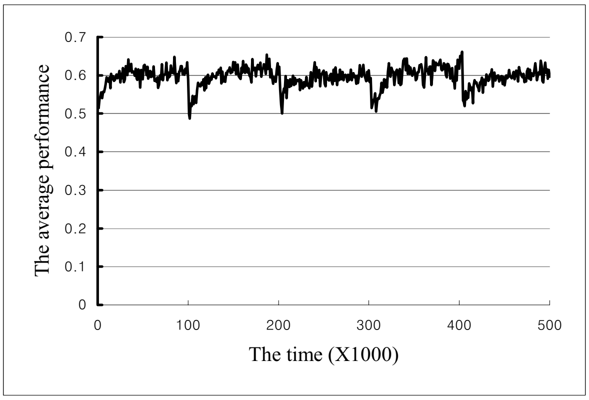

Fig. 5. The average performance of the system when the type of tasks is periodically changed.

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

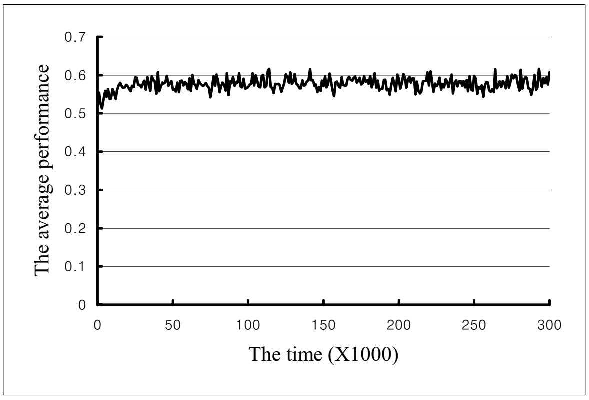

Fig. 6. Average speed of operation in four tasks.

5. CONCLUSION

In the last, the 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 more needed at the future, when the autonomous robots become 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 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 don't communicate the information of environment to other robot. However, they decide proper action from its environment. The performance of the system is not good but shows that the system adapts to unknown environment and changes. Consequently, embodying adaptive DARS, which is based on AIN and XCS, is possible. However, 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] N. K. Jerne, "Idiotopic Network and Other Preconceived Idias," *Immunological Rev*., vol. 79, pp. 5-24, 1984.

[3] D.W. Lee, and K.B Sim, "Artificial Immune Network-based Cooperative Control in Collective Autonomous Mobile Robots," *Proceedings of the 6th IEEE International Workshop on ROBOT AND HUMAN COMMUNICATION(RO-MAN)*, pp. 58-63, 1997. 9. 29 – 10. 1

[4] D.W. Lee, H.B. Jun, and K.B. Sim, "Artificial Immune System for Realization of Cooperative Strategies and Group Behavior in Collective Autonomous Mobile Robots", *Proceedings of The 4th International Symposium on Artificial Life and Robotics*, vol. 1, pp. 232-235, 1999. 1. 20

[5] D.W. Lee, K.B. Sim, "Cooperative Behavior of Collective Autonomous Mobile Robots Based on Online Learning and Evolution," *Proceedings of 7th International Conference on Neural Information Processing(ICONIP 2000)* , 2000. 11. 14-18.

[6] S.W. Wilson, "Classifier Fitness Based on Accuracy", *Evolutionary Computation*, Vol. 3, No. 2, pp. 149-175, 1995.

[7] T. Kovacs, "Evolving Optimal Populations with XCS Classifier Systems", *MSc. Dissertation, Univ. of Birmingham, UK*, 1996, 10.

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

[9] M. Butz, S.W. Wilson, "An Algorithmic Description of XCS", *Soft Computing*, Vol. 1996, pp. 256-273, 2001.