Fifth International Workshop on Computational Intelligence & Applications
IEEE SMC Hiroshima Chapter, Hiroshima University, Japan, November 10, 11 & 12, 2009

An Evolutionary Multi-Objective Optimization-Based Constructive Method for Learning Classifier Systems Adjusting to Non-Markov Environments

Keita Moriwake, Hideki Katagiri, Ichiro Nishizaki and Tomohiro Hayashida
Graduate School of Engineering, Hiroshima University
Kagamiyama 1-4-1, Higashi-Hiroshima City, Hiroshima, 739-8527 Japan
email:{m094032,katagiri-h,nisizaki,hayashida}@hiroshima-u.ac.jp

***Abstract*—Learning Classifier Systems (LCSs) are rule-based systems that automatically build their rule set so as to get optimal policies through evolutionary processes. This paper considers an evolutionary multi-objective optimization-based constructive method for LCSs that adjust to non-Markov environments. Our goal is to construct a XCSMH (eXtended Classifier System - Memory Hierarchic) that can obtain not only optimal policies but also highly generalized rule sets. Results of numerical experiments show that the proposed method is superior to an existing method with respect to the generality of the obtained rule sets.**

I. Introduction

LCSs are a set of classifiers which consists of condition part, action part and fitness. LCSs are input perceptual state into and learn optimal outputs by genetic algorithms. Each classifier represents a IF-THEN rule whose usefulness is evaluated by the fitness calculated based on its condition and action parts. LCSs have been applied to various fields, for example, clustering technique, automated discrimination problem, robotic behavioral control and so on.

LCSs are not black box systems such as neural-network and can extract essential and general rules from a set of classifiers that can derive optimal policies. It is apparent that the more compact and more essential the rule extracted from a set of classifiers are, the more useful the systems are. From this viewpoint, Mansilla *et al* [1]. have proposed LCSs based on evolutionary multi-objective optimization method focusing on trade-off between generality and accuracy of the rules. However, their LSCs have two restrictions: One is that their system employs supervised learning with training data, where optimal policies are known in advance so that their system cannot deal with problems which do not have a set of training data, for example, sequential learning problems. The other is that their system cannot deal with problems in non-Markov environments where each of states in systems depends on not only a current situation but also past situations, although LCSs for non-Markov environments becomes more and more important in order to solve the real-world complex problems.

Holland [4] firstly introduced LCSs in 1975, and Wilson [5] proposed ZCS that includes a concept of fitness which is a degree of expected reward. Since ZCS has a drawback that the system eliminates a classifier with a few predicted rewards even if it includes an optimal policy, XCS [6] proposed by Wilson improves such a drawback of ZCS by regarding the fitness of classifier in XCS as the accuracy of the rule of classifier.

Classifiers of XCS have not only fitness but also the expected reward p: The fitness of XCS is computed not by the amount of expected rewards p but by the error between the expected rewards p and the real rewards. In order to solve the problem in Non-Markov environments that XCS cannot deal, XCSMH (eXtended Classifier System - Memory Hierarchic) was proposed by Lanzi [2] through the introduction of internal states representing internal registers.

In the subsequent sections, after describing some previous studies and their drawbacks, we construct a XCSMH-based LCS that can obtain optimal policies together with highly generalized rule sets. Furthermore, through results of numerical experiments, we show that the proposed method is superior to an existing method with respect to the generality of the obtained rule sets.

II. Learning classifier systems with multi-objective optimization in Markov environment

Fig. 1 shows the outline of ZCS. In LCSs, a condition part of each classifier is expressed by a set of value taken from $\{0, 1, \sharp\}$ that corresponds to situations obtained by observing environments. The notation $\{\ \sharp\ \}$, called $don't\ care$, means that it can take either $\{\ 0\ \}$ or $\{\ 1\ \}$ of situation. For example, $\{\ 0\ 0\ 1\ \sharp\ \}$ corresponds two possible patterns of situation, $\{$ 0010 $\}$ and $\{$ 0011 $\}$.

First, LCSs conduct the operation called ”matching”, which checks whether there exist classifiers having conditions matching a given situation, and if the answer is yes, such classifiers are called ”firing” classifiers. Next, LCSs select an action to the environment from a set of firing classifiers. Finally, the fitness which evaluates availability of rule is updated by the rewards given by environments that is based on whether

the action given from LCSs to environment is accurate, and evolutionary learning by running genetic algorithm based on fitness is implemented.

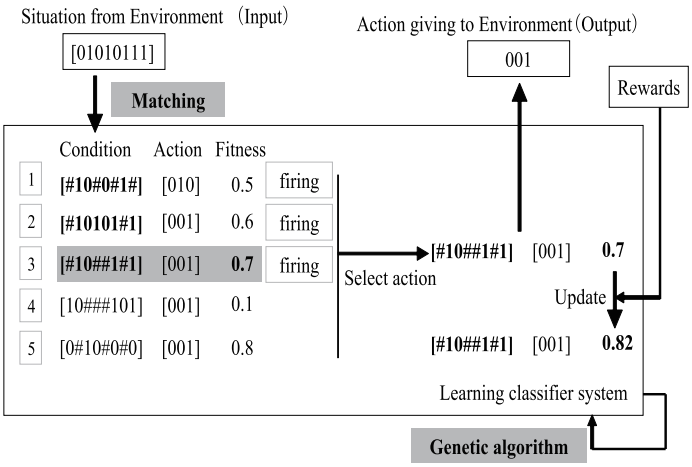

Fig. 1. Outline of LCSs

The classifier which have a number of patterns matching situations given from environments has high generality that is presented with more compact conditions. Although the accuracy rules which derive optimal policy is useful, more useful rules are ones with not only highly accuracy but also high generality. However, there is some trade-off between generality and accuracy, which means that higher generality may lead to make it low accuracy, or it may be necessary to lower generality to improve accuracy. Therefore, a set of rules which lead to optimal policies generally exist around a Pareto-optimal frontier set where both the generality and accuracy of rules are high. From this viewpoint, Mansilla *et al* [1]. have incorporated the ideas of evolutionary multi-objective optimization focusing trade-off between generality and accuracy into XCS through the introduction of the following two objective functions:

$$F_{G_{man}} = \frac{\text{the number of training data covered by classifiers}}{\text{the number of training data}}$$

$$F_{A_{man}} = \frac{\text{the number of correctly classified training data}}{\text{the number of training data covered by classifiers}}$$

where $F_{G_{man}}$ and $F_{A_{man}}$ represent the generality and accuracy of classifier, respectively.

Since their system is constructed based on supervised learning using training data in which an optimal policy for each of situations is known, it cannot deal with problems without training data such as sequential learning problems. Moreover, because this system is based on XCS, it also cannot deal with problems in non-Markov environments. Therefore, in this paper, we use XCSMH instead of XCS and consider a new evolutionary multi-objective optimization-based constructive method for LCSs adjusting to non-Markov environments by introducing a concept of generality that can be used without training data.

III. New learning classifier systems based on multi-objective optimization and XCSMH

As observed in maze running tasks [2] known as benchmark problems of LCSs for testing the capability of the systems, in non-Markov environments, there exist generally alias situations for which only the information of current situation is not sufficient to get an optimal policy. For example, there are two situations which have the same perceptual information, while their optimal actions are different.

While XCS has only the external condition which corresponds to a situation from environments and external action, XCSMH proposed by Lanzi [2] has not only external condition and action but also the internal condition which corresponds to internal state and the internal action which modifies internal state to each classifiers. Through the introduction of such internal condition and action, XCSMH can derive optimal policies by distinguishing alias situations as different internal states employing internal actions.

The scheme of MOXCSMH (Multi-Objective XCSMH) which is proposed in this paper is depicted as shown in Fig. 2.

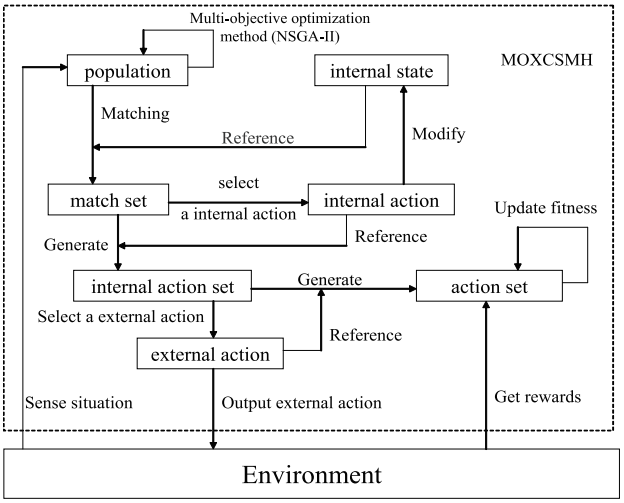

Fig. 2. The scheme of MOXCSMH

The condition of classifiers of MOXCSMH is composed of the external condition corresponding to the situation of environments and the internal conditions corresponding to the internal state. On the assumption that the number of $don't\ care$ of the external condition is $n_{ext\sharp}$, and that of the internal condition is $n_{int\sharp}$, we can calculate the number of patterns of situation as $2^{n_{ext\sharp}} \times 2^{n_{int\sharp}}$. Accordingly, we define the objective function of generality F_G as follows:

$$F_G = 2^{n_{ext\sharp}} \times 2^{n_{int\sharp}} \quad (1)$$

The fitness of MOXCSMH identical to XCS is based on the accuracy of rule of the classifier. The classifier have not only the fitness but also the following value p which represents predictive rewards given from environments and is updated using learning rate β and real rewards ρ:

$$p \leftarrow p + \beta(\rho - p)$$

The fitness of MOXCSMH have the relation of trade-off between generality and accuracy defined objective function of accuracy F_A as the fitness of MOXCSMH. The trade-off between generality and accuracy can be represented by using these objective functions F_G and F_A. In order to get a set of useful rules with high generality and accuracy, we incorporate NSGA-II [3] proposed by Deb *et al*, which is one of the effective multi-objective optimization methods, into MOXCSMH.

A. The algorithm of MOXCSMH

We show the algorithm of the proposed method MOXCSMH as below. Assume that current time step is t, the following flow of treatment is one period and experiment in established learning period and test period.

Step 1 [Generation of match sets]
Generate the match set composed of firing classifiers by implement of matching between the external condition and the situation from environment and between the internal condition and the internal state.

Step 2 [Selection of internal actions]
Select the internal action from the classifiers belonging the match set. If the number of actions present in the match set is lower than the constant value, create a new classifier, and add it to population and the match set for augmenting the number of the actions present in the match set. Generate the internal action set from the classifiers which have the selected internal action in the match set.

Step 3 [Selection of external actions]
Select the external action from the classifiers belonging the internal action set. If the number of actions present in the internal action set is lower than the constant value, create a new classifier, and add it to population and the internal action set for augmenting the number of the actions present in the internal action set. Generate the action set from the classifiers which have the selected internal action in the match set.

Step 4 [Taking actions and updating the fitness values]
Execute the selected action and internal action and update the fitness of classifiers belonging action set based on the rewards from environment and expected rewards.

Step 5 [Evolutionary multi-objective optimization]
Generate new classifier by applying crossover and mutation, and implement multi-objective optimization method about generality and accuracy using NSGA-II for the newly created population.

Step 6 [At-end condition]
If the situation transfer to the end status, end trials. Otherwise, return to step 1 and $t \leftarrow t + 1$.

The flowchart of MOXCSMH is depicted as shown in Fig. 3.

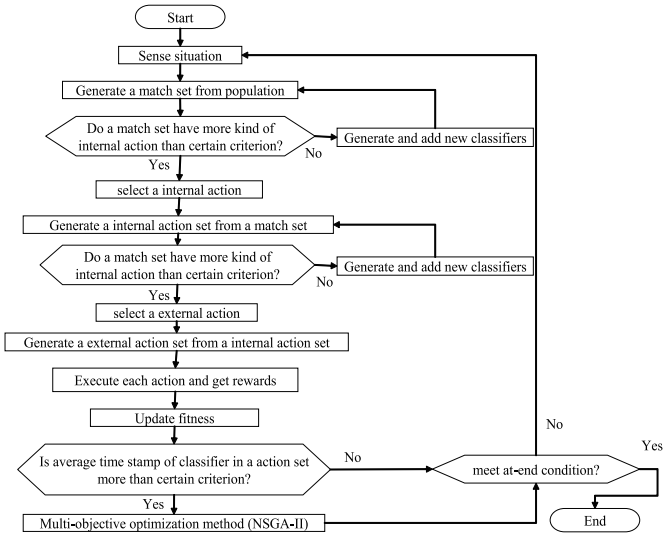

Fig. 3. The flowchart of MOXCSMH

IV. Analysis of the capability of MOXCSMH

This section implements performance comparison with respect to learning capability and a rule of generality between XCSMH which is an existing method and MOXCSMH which is the proposal method with the use of maze running task in Non-Markov environment, and analysis of the result.

A. Benchmark problems

The benchmark problems are the maze running tasks in Non-Markov problem depicted in Fig. 4 called $woods101\frac{1}{2}$. There are grids in which each cell represents an obstacle, free, or a goal "G", and the agent has to learn the shortest path to the goal state. The policy which lead to the shortest path is optimal. The agent senses the environment by means of Boolean sensors that report features of the contents of each of eight adjacent cells, and can move into any adjacent cell that is free. This problem is a Non-Markov problem which cannot lead to optimal policy by only using current situation, because all the cells labeled "A" in Fig. 4 are alias states whose situations are regarded as the same ones, while the optimal action for each of situations is different from each other. The start position is randomly selected from "S" cells, the agent proceeds toward a goal "G" through alias "A"cells. The optimal policies for this problem are depicted by arrows in Fig. 5, and its average time steps is 4. It is possible to evaluate the accuracy F_A of classifier by XCSMH [2].

The experiment consists of learning period and test period. One period is defined as one process of agents starting from the "S" cell to arriving at the "G" cell. The learning period is from

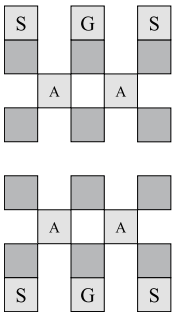

Fig. 4. $woods101\frac{1}{2}$

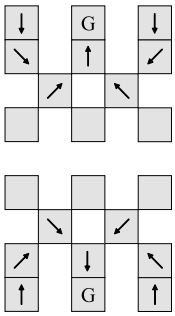

Fig. 5. The optimal policy

1 period to 27000 period and the test period is from 27001 period to 30000 period. This paramerter of period is enough to evaluate capability of XCSMH and MOXCSMH [2]. In the learning period, the agent selects not only the action which have the highest expected rewards but also the action selected randomly or by a roulette wheel mechanism. Moreover, GA and NSGA-II are applied to only learning period. On the other hand, in the test period, the agent selects only the action with the highest expected rewards, and GA and NSGA-II are not applied during the period. The limit of population size is 30000 and learning rate is 0.2 at each trial.

B. The result about the learning capability

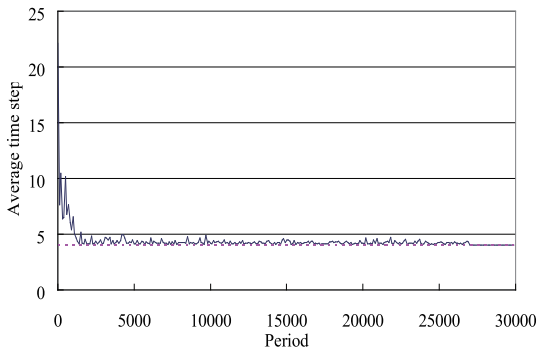

Fig. 6. The process of average time steps in XCSMH

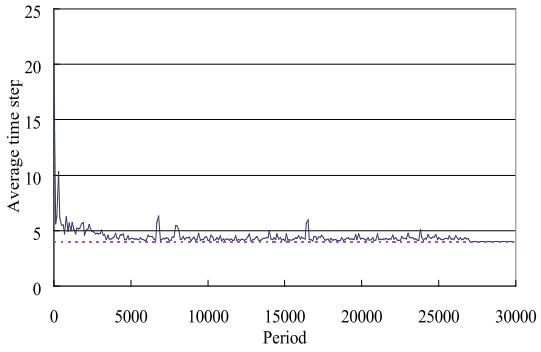

Fig. 7. The process of average time steps in MOXCSMH

Figs. 6 and 7 depict the process of average time step of 100 periods from the start "S" cell to the goal "G" cell. In learning period, because the agent selects action randomly a few times, the time step is larger than 4 which is optimal. However, the average time step converge the optimal one at both XCSMH and MOXCSMH, thus the result indicates that the agent can learn the optimal policy as shown in Fig. 5.

Accordingly, we can conclude that the learning capability of MOXCSMH is as well as XCSMH's in $woods101\frac{1}{2}$.

C. Analysis on generality of the rules

1) Number of firing situations: The more number of $don't\ care$ in the part of condition the classifiers have, the more number of situations which match the condition the classifiers have. The generality of classifiers is defined by the compactness of conditions, thus the classifier with high generality can derive an action by utilizing fewer input information. Fig. 8 depicts the comparison of the numbers of firing situations between XCSMH and MOXCSMH, which are calculated by Eq. 1.

Each number of horizontal axis correspond to the cell number depicted in Fig. 9, and each value of vertical axis is the average number of ten trials of firing situations in which each cell is passed through in optimal polices. The dashed lines are the average of all the values that correspond to the cell numbers.

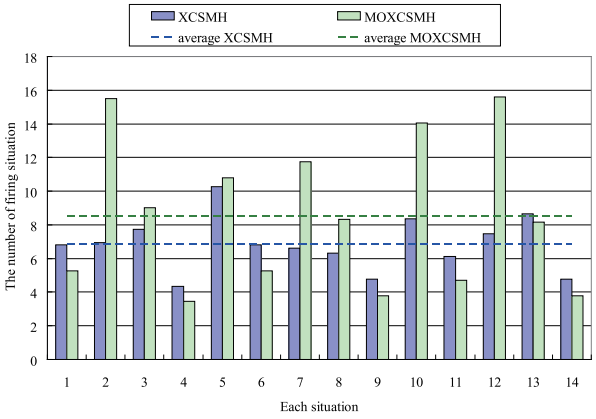

Fig. 8. Comparison of the number of firing situations

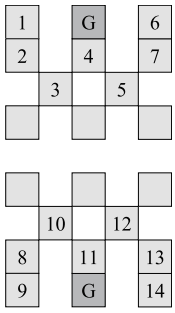

Fig. 9. The cell passed in optimal policy

Fig. 8 shows that the number of firing situations of MOXCSMH is almost larger than those of XCSMH, and the average number of firing situations is higher. XCSMH implements the

subsumption process which subsume classifiers that have low generality to the classifier have the highest generality, thus the classifier with high generality tend to be generated, but subsumption process only considers the external condition. Several positions have nearly identical value of the number of firing situation, because when learning about accuracy ends, learning by GA or NSGA-II ends at the same time, even if learning about generality has not finished yet.

2) Extraction of rules: This subsection devotes to analyzing extracted rules from a set of classifiers which derive optimal policies. For example, Fig. 10 represents the condition which the classifier firing at the cell at number 12 uses. In this representation, ”0” is an obstacle and ”1” is a free cell.

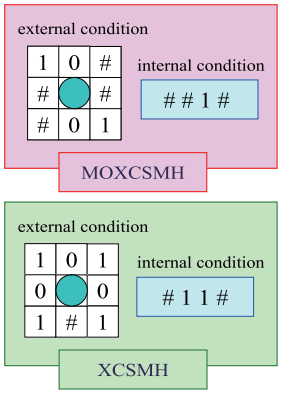

Fig. 10. Extracted rule in the cell at number 12

The agent has to distinguish the situation of the cell at number 12 from others that is numbers 1, 2, 4, 6, 7, 8, 9, 11, 13 and 14. Fig. 11 represents the distinction of situations between number 12 and others. As shown in Fig. 11, the most compact condition exist three patterns and can be depicted as shown in Fig. 12.

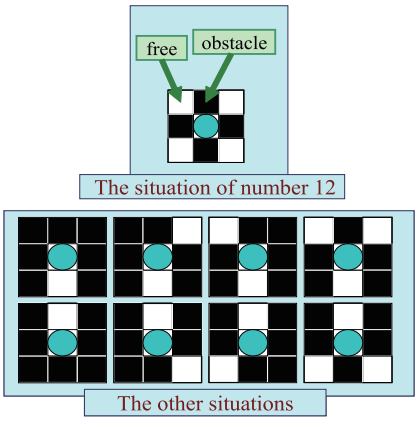

Fig. 11. Comparison of situation between the cell at number 12 and others

It is observed from Figs. 10 and 12 that the extracted condition of MOXCSMH subsumes a simple rule that is ”if the upper cell and the lower cell are obstacle”. By contrast, the extracted condition of XCSMH subsumes two simple rules that are ”if the upper right cell and the lower right cell is free”

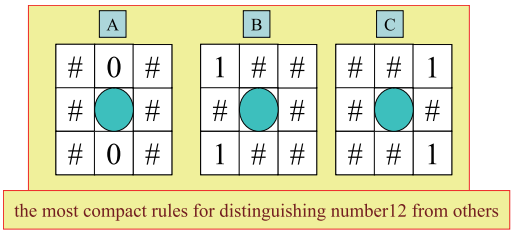

Fig. 12. The most compact condition which distinguish the cell at number 12 from others

and ”if the upper left cell and the lower left cell is free”. This result indicates that MOXCSMH can generate classifiers with higher generality than XCSMH, although neither MOXCSMH nor XCSMH does not extract the most compact condition.

The reason why MOXCSMH and XCSMH cannot learn the most compact condition is that the procedure of GA and NSGA-II is not implemented when classifiers have a relatively high accuracy and that the at-end condition does not depend on learning about generality. It should be noted here that as shown in Fig. 3, the procedure of the multi-objective optimization method based on NSGA-II stops after exploring for rules with high accuracy even if the exploration for getting rules with high generality is not completed because this procedure can be implemented only if the value of time stamp in action sets is larger than a certain constant. Therefore, for generating more compact conditions, a new termination condition for leaning should be constructed based not only on accuracy but also on generality.

V. Conclusion

In this paper, we have proposed a multi-objective classifier system in non-Markov environments, which takes account of not only the accuracy but also the generality of the rules. We have defined the objective function of generality as the number of firing situations considering the external condition and the internal condition of the classifier. The experimental results have shown that MOXCSMH can obtain a set of rules with higher generality than XCSMH, and can learn an optimal policy in non-Markov environments as well as XCSMH.

In the future, we will consider a more suitable multi-objective optimization method for classifier systems.

References

[1] E. Bernado-Mansilla, X. Llora, I. Traus, “Multiobjective Learning Classifier Systems:An Overview”, IlliGAL Report, No.2005020, 2005

[2] P. Lanzi, S. Wilson, “Toward optimal classifier system performance in non-Markov environments”, Evolutionary Computation, 8, pp. 393–418, 2000

[3] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II”, Evolutionary Computation, 6, pp. 182–197, 2002.

[4] J. Holland, “Adaptation in Natural and Artificial Systems: An Introductry Analysis with Applications to Biology”, *Control, and Artificial Intelligence*, University of Michigan Press, Ann Arbor, MI., 1975.

[5] S. Wilson, “ZCS, a Zeroth level Classifier System.”, *Evolutionary Computation*, 2(1), pp. 1–18, 1994.

[6] S. Wilson, “Classifier Fitness Based on Accuracy”, *Evolutionary Computation*, 2(1), pp. 149–175, 1995.