An Evolutionary Multiobjective Optimization-Based Learning Classifier System in Non-Markov Environments

Hideki Katagiri∗, Ichiro Nishizaki∗, Tomohiro Hayashida∗ and Keita Moriwake∗

∗Graduate School of Engineering Hiroshima University

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 multiobjective optimization-based method for constructing LCSs that adjust to non-Markov environments. Our goal is to construct an 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.

Keywords: Learning classifier system, evolutionary multiobjective optimization, non-Markov environment, accuracy and generality of the rule

1 Introduction

Learning classifier systems (LCSs) are rule-based systems which consist of a set of classifiers together with the concepts of condition part, action part, fitness and so on, originally introduced by Holland [4] in 1975. Each classifier represents an IF-THEN rule whose usefulness is evaluated by the fitness calculated based on its condition and action parts. LCSs are powerful tools for solving many real-world problems, for example, classification of biomedical datasets such as liver cancer data, diabetes data [3], breast cancer data [6] and gene expression data [8, 11].

Wilson [9] proposed ZCS that includes a concept of expected rewards. Since ZCS has a drawback that the system eliminates a classifier with a small amount of predicted rewards even if it includes an optimal policy, Wilson [10] proposed XCS by regarding the fitness of classifier in XCS as the accuracy of the rule of classifier.

Classifiers of XCS have the fitness and the expected reward: the fitness of XCS is computed not by the amount of expected rewards but by the error between the expected rewards 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 [7] through the introduction of internal states representing internal registers.

In general, LCSs are not black box systems such as neural networks, and can extract essential and general rules for solving the problems from a set of obtained classifiers. 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 multiobjective optimization method focusing on trade-off between generality and accuracy of the rules. However, their LSC has two drawbacks: One is that their system employs supervised learning with training data, the other is that their system cannot deal with problems under non-Markov environments where the next situation depends on not only a current situation but also past situations.

In this paper, after describing some previous studies and their drawbacks, we construct an XCSMH-based LCS that can obtain optimal policies together with highly generalized rule sets. To be more precise, there are two main contributions of this paper; one is to develop the new procedure of generating a set of rules with high generality, called EORHG (Explorations for Obtaining Rules with High-Generality), and the other is to propose an extended evolutionary multiobjective optimization method. By incorporating the two proposed methods into our model, we construct a novel LCS under non-Markov environments.

Through the results of numerical experiments, we show that the proposed method is superior to an existing method with respect to the availability of the rules obtained from the classifiers.
2 Existing learning classifier systems

2.1 Outline of learning classifier systems

In this section, as a representative of LCSs, we review the mechanism of ZCS, as shown in Fig. 1. In LCSs, a condition part of each classifier is expressed by a set of value taken from \{0, 1, ♯\} that corresponds to situations of environments. The notation \{♯\}, called don’t care, means that it can take either \{0\} or \{1\} of situation. For example, \{0 0 1 ♯\} corresponds to 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. 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 the availability of the rule is updated by the rewards given by environments, and the amounts of rewards are calculated based on the degree of accuracy of the selected action of LCSs, and an evolutionary learning process using genetic algorithms is implemented.

\[ F_{G_{\text{man}}} = \frac{\text{Number of training data covered by classifiers}}{\text{Number of training data}} \]
\[ F_{A_{\text{man}}} = \frac{\text{Number of correctly classified training data}}{\text{Number of training data covered by classifiers}} \]

where \(F_{G_{\text{man}}}\) and \(F_{A_{\text{man}}}\) represent the degrees of generality and accuracy of classifiers, respectively.

Since their system is constructed based on supervised learning using training data in which an optimal policy for each of situations is known in advance, it cannot deal with problems without training data such as sequential learning problems. Moreover, since this system is based on XCS, it cannot also deal with problems in non-Markov environments.

2.2 Multiobjective optimization-based learning classifier systems

In general, when the classifier has a number of patterns matching situations of environments, it can be regarded as a classifier with high generality. Although it is meaningful to get some accurate rules which derive optimal policy, it is more important to obtain rules with not only high accuracy but also high generality. Here, it should be noted here that there is generally some trade-off between generality and accuracy, which means that the higher generality of rules may lead to the low accuracy of the rules. Therefore, a set of rules which lead to optimal policies generally exist around a Pareto-optimal frontier where both the degree of generality and that of accuracy of rules are simultaneously high. From this viewpoint, Mansilla et al. [1] incorporated the ideas of evolutionary multiobjective optimization focusing trade-off between generality and accuracy into XCS through the introduction of the following two objective functions:

2.3 Learning classifier systems in non-Markov environments

As observed in maze running tasks [7] which are well 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.

Although XCS has only the external condition and action which are related to a situation perceived from environments, XCSMH proposed by Lanzi [7] has not only the external condition and action but also the internal condition and action; the internal condition corresponds to internal state, and the internal action modifies the internal state of each classifier. Through the introduction of such an internal condition and action, XCSMH can derive optimal policies by distinguishing alias situations as different internal states employing internal actions.

While XCSMH is a useful tool for dealing with problems in non-Markov environments, it does not take account of the generality of the rules derived from the obtained classifiers.

Under these circumstances, in this paper, we consider a new learning classifier system by incorporating a revised evolutionary multiobjective optimization-based method into the XCSMH.
3 Evolutionary multiobjective optimization-based learning classifier system in non-Markov environments

In this section, we propose an Evolutionary Multi-objective Optimization-based XCSMH (EMOXCSMH) for handling problems in non-Markov environments. The scheme of EMOXCSMH is depicted as shown in Fig. 2.

3.1 Index function for measuring the generality of the classifiers

The condition of classifiers of EMOXCSMH is composed of the external conditions corresponding to the situation of environments and the internal conditions corresponding to the internal states. Let \( n_{ext} \) and \( n_{int} \) be the number of don't care of the external condition and that of the internal condition, respectively. Then, the number of patterns of situation is expressed as \( 2^{n_{ext}} \times 2^{n_{int}} \). Accordingly, we define the index function \( F_G \) for measuring the generality of the rules as follows:

\[
F_G = 2^{n_{ext}} \times 2^{n_{int}}
\]  
(1)

3.2 Index function for measuring the accuracy of the classifiers

Next, we define the index function for measuring the accuracy of the classifier. For constructing a proper index, it is important to take account of the trade-off relationship between the number of states on which a classifier fires and the fitness value \( c_i.F \) of the classifier: if the degree of generality of the condition part in the classifier is too high, such a classifier fires on a number of states, which leads to the variability of the obtained reward. This means that the predictive reward \( c_i.p \) is also variable, and that the error \( c_i.e \) becomes large, which results in the low fitness value \( c_i.F \).

One of possible forms of index function for measuring the degree of accuracy of the classifier is \( c_i.F \times c_i.as \), where \( c_i.as \) is the action set size of classifier \( c_i \). However, this form is not appropriate because it does not consider the critical issue of XCSMH: even if the external condition of a classifier in XCSMH includes too many don't care, then the corresponding classifier may derive an optimal policy. In other words, even if the system can solve the problem, we may not obtain any understandable rules or knowledge for solving the problem.

In order to get classifiers which can provide useful rules, it is necessary to reduce the evaluation of such classifiers that have too high degrees of generality. From this point of view, we define a new index function \( F_A \) for measuring the degree of accuracy by using the number of firing states, denoted by \( n_{firing} \), as follows:

\[
F_A = \frac{c_i.F \times c_i.as}{n_{firing}}
\]  
(2)

3.3 New evolutionary multiobjective optimization method for learning classifier systems

Although NSGA-II [2] proposed by Deb et al is a powerful tool for obtaining a set of Pareto optimal solutions, if the original NSGA-II is directly applied to obtaining a desirable classifier as the solution method for two objective problem, there are critical issues about the low learning speed and the instability of learning convergence.

Therefore, in this paper, we propose a revised NSGA-II by incorporating new ideas of "active removal procedure" and "elite preservation strategy".

3.3.1 Active removal procedure

In the NSGA-II, if the sum of the original classifiers and the new classifiers, which are generated by crossover and mutation operations, exceeds a certain upper limit \( N \), some classifiers whose degrees of usefulness are lower are removed from the population. In other words, if the sum does not exceed the upper limit \( N \), then the procedure of multiobjective optimization is not implemented, and the newly generated classifiers are just added to the population in a way similar to GA.
Fig. 2: Framework of the proposed EMOXCSMH

Fig. 4 shows the individual removal procedure implemented in the original NSGA-II: the first step is that one individual which is newly generated is added to the population, and the second one is that one individual whose degree of usefulness is the lowest is removed from the population through non-dominated solution ranking and crowding distance sorting. Thus, the efficiency of multiobjective optimization depends on the number of the individuals which are newly added to the population.

In order to tackle such a difficulty, we consider a new concept of active removal. Fig. 5 shows the active removal procedure in the revised NSGA-II when the number \( N_{ar} \) of individuals to be removed is \( N_{ar} = 2 \). With this process, a number of individuals whose objective functions are bad can be actively removed from the population, which leads to the intensification of exploring the solution space.

3.3.2 Elite preservation strategy

In the original NSGA-II, the individuals to be removed are chosen dependent on the objective function values. If we directly apply the original one to learning the classifiers, then there is a possibility that some classifiers which are necessary for the good learning in the future are removed. For avoiding such a difficulty, we consider a new concept of elite preservation strategy, which regards the individual whose has the highest degree of accuracy (\( F_A \)) in the action set as the elite individual. The elite individual is categorized as rank 1, which means that the elite individual is regarded as the most important individual which cannot be removed.
3.4 Exploration for obtaining rules with high generality (EORHG)

The aim of the original XCSMH is to obtain a set of accurate rules in non-Markov environments, and it does not focus on the generality of the obtained rules. In order to obtain rules with high generality, we consider the process of Exploration for Obtaining Rules with High Generality (EORHG). To be more specific, the procedure adds a new classifier which is constructed by replacing 0 or 1 by "don’t care" in the condition part of the high-accuracy classifier.

Let \( cl.exp \) and \( cl.\epsilon \) be the experience point and the error of classifier, respectively. For the classifiers in the action set, if the average of \( cl.exp \) is greater than or equal to \( \theta_{sub} \) and the average of \( cl.\epsilon \) is less than \( \epsilon_0 \), we perform the following procedure which consists of three steps at probability \( P_{LG} \): 1) find the classifier in the action set which has the largest \( F_A \) in Eq. 2, denoted by \( cl_{maxF_A} \), 2) generate a new classifier \( cl_{LG} \) by replacing one value not being "don’t care" in the condition part of \( cl_{maxF_A} \) with "don’t care", 3) add the classifier \( cl_{LG} \) to the population by setting \( cl_{LG}.exp = 0 \) and \( cl_{LG}.F = cl_{maxF_A}.F \times 0.1 \). The values of parameters except for \( cl_{LG}.exp \) and \( cl_{LG}.F \) are set as the same value of \( cl_{maxF_A} \).

3.5 Algorithm of EMOXCSMH

Let \( t \) be the current time step. Then, the following is the algorithm of the proposed EMOXCSMH:

**Step 1** [Generation of the match sets]

Generate the match set composed of firing classifiers by implementing the matching between the external condition and the situation from environment as well as the matching between the internal condition and the internal state.

**Step 2** [Selection of internal actions]

Select an internal action from the classifiers belonging to the match set. If the number of actions in the match set is lower than a given constant value, create a new classifier, and add it to the population and the match set for augmenting the number of the actions 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 to the internal action set. If the number of actions in the internal action set is lower than a given constant value, create a new classifier, and add it to the population and the internal action set for augmenting the number of the actions 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 values of classifiers belonging to the action set based on the rewards from environments and expected rewards.

**Step 5** [Evolutionary multiobjective optimization]

Generate new classifiers by applying crossover and mutation, and implement the evolutionary multiobjective optimization method using the revised NSGA-II for the newly created population.

**Step 6** [Explorations for obtaining the rules with high degrees of generality]

If the iteration number is within the period of exploring the rules with high generality (EORHG period), the average of \( cl.exp \) is not less than \( \theta_{sub} \), and the average of \( cl.\epsilon \) is less than \( \epsilon_0 \), then perform the process of exploring the rules with high generality at probability \( P_{LG} \).

**Step 7** [Termination condition]

If the situation meets the termination condition, then stop. Otherwise, return to step 1 and \( t \leftarrow t + 1 \).

In Step 5, we use the concept of "time stamp" which is a variable for recording the last period when a GA is implemented. The revised NSGA-II is implemented if it holds that \( ts_{av.act} - t > \theta_{GA} \), where \( ts_{av.act} \) is the average time stamp. The flowchart of EMOXCSMH is depicted as shown in Fig. 6.

4 Experimental results

This section devotes to comparing the performances of the proposed EMOXCSH and the XCSMH with respect to the learning capability and the generality of the classifiers through maze running tasks in non-Markov environments.

4.1 Benchmark problems

As a benchmark instance problem, we use the maze running tasks in non-Markov problem depicted in Fig. 7, called woods1015. There are grids in which each cell represents an obstacle, free space, or a goal, and the agent has to learn the shortest path to the goal. The policy which derives the shortest path is optimal. The agent can move into any adjacent cell that is free by sensing the environment through Boolean sensors that realize what cells the eight adjacent cells are. It should be noted here that an optimal policy cannot be obtained by only using the current situation, because all the cells labeled "A" in Fig. 7 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. 8, and its average time step is 4.

Figure 7: woods101 1/2

Figure 8: Optimal policy

In the experiments, we use the following parameter values.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training period</td>
<td>0 - 17000</td>
</tr>
<tr>
<td>Test period</td>
<td>17001 - 19999</td>
</tr>
<tr>
<td>EORHG period</td>
<td>5000 - 8000</td>
</tr>
<tr>
<td>Internal state number</td>
<td>4 bit</td>
</tr>
<tr>
<td>$P_{LG}$</td>
<td>0.2</td>
</tr>
<tr>
<td>$\theta_{sub}$</td>
<td>20</td>
</tr>
<tr>
<td>$\epsilon_0$</td>
<td>10.0</td>
</tr>
</tbody>
</table>

4.2 Experimental results on learning ability

Figs. 9 depicts the average time step for every 100 periods from the start "S" cell to the goal "G" cell. In the learning period, because the agent selects actions randomly a few times, the time step is larger than 4 which is optimal. On the other hand, in the test period, the average time step converges to the optimal one, and thus the result indicates that the agents can learn the optimal policy. However, as discussed before, the original XCSMH does not consider the generality of the rules. In fact, as will be discussed later, the rules obtained by XCSMH are not useful. Before showing the performance of the proposed EMOXCSMH, as an attempt to get the rules with high generality, we investigate the model in which the procedure of EORHG is directly incorporated into the original XCSMH, called XCSMH with EORHG.

Fig. 10 shows the average time step for XCSMH with EORHG in which the upper limit of individuals is 10000. From Fig. 10, it is observed that the average time step increases rapidly during the process of EORHG. This is caused by a decline in training ability of XCSMH due to a lack of classifiers for exploring a wide range of exploration space. In this experiment, the number of states is 22, the number of external actions is 8, and the number of internal actions is 3^4. Then, the number of classifier necessary to cover the
The whole training space is $22 \times 8 \times 3^4 = 14256$. Since 14256 is apparently larger than the upper limit of individuals, 10000, there is a high possibility that the XCSMH cannot be well trained. When the XCSMH is not well trained, almost all the individuals in the population are covered with the classifiers with $c_{l,a} = 1$, which leads to the removal of the classifiers necessary for the good training as a result of the random choice of classifiers to be removed. In addition, in the test period, we observe that the obtained classifier cannot derive an optimal policy, although the average time step is almost convergent. Next, we conduct the experiment in which the upper limit of individuals is 20000. It is observed from Fig. 11 that the training of the XCSMH is successful so that an optimal policy can be derived.

Fig. 12 depicts the average time step for reaching the goal when EMOXCSMH is applied. In the system, the upper limit of individuals is 10000. Although the values in the training period are greater than 4 steps which correspond to the optimal time step, the values in the test periods exactly approach 4 steps, which means that the optimal rules represented in Fig. 8 can be derived. Due to the effect of active removal in EMOXCSMH, unlike the XCSMH with EORHG, an optimal policy can be derived in spite of small size of the upper limit of individuals (10000). Hence, the proposed EMOXCSMH can obtain an optimal policy with shorter computational time than XCSMH with EORHG.

4.3 Experimental results on the generality of the obtained rules

Next, we focus on $n_{firing}$, the number of firing states of classifiers in the action set, to examine the generality of the obtained classifiers. It should be noted here that the maximum of $n_{firing}$ equals to the number of states necessary for describing the problem to be solved, and that the minimum is always 1. In the case of woods101 1 1 2, the number of states necessary for describing the problem to be solved is 22. Figs. 13 and 14 depict the number of firing states of the original XCSMH and XCSMH with EORHG, respectively. Fig. 13 shows that the generality of the rules obtained by the original XCSMH is low. It is observed from Fig. 14 that the values rapidly increases from 5000 period (the starting period of EORHG), which means that the training of XCSMH is well conducted. However, there is a critical issue on the obtained rules; according to Fig. 14, there is a possibility that a classifier whose firing state is 22 can derive an optimal policy. It should
be noted here that all the elements in the external part are "don’t care" in a classifier whose firing state is 22, which means no useful rule can be derived from such a classifier in spite of its optimality.

Figure 13: Number of firing states iXCSMH [7])

Figure 14: Number of firing states iXCSMH with EORHG)

Figure 15: Number of firing states (EMOXCSMH)

Fig. 15 shows the number of firing states of EMOXCSMH. Unlike Fig. 14, as the evolution process proceeds, the maximum number of firing states is getting lower from the theoretical maximum number 22, which means that the proposed multiobjective optimization method can properly remove the classifiers with too high degrees of generality from the population. Actually, unlike the XCSMH with EORHG, there is no classifier in which all the elements in the external state are don’t care. In the next subsection, we discuss in detail the generality and availability of the rules extracted from the obtained classifiers.

4.4 Rule extraction

If a set of classifiers can derive an optimal policy, each of classifiers in such a set represents a useful rule. In this subsection, for woods1012, we extract a set of rules from a set of classifiers of three types of LCS; a) XCSMH [7], b) XCSMH with EORHG and c) EMOXCSMH. For extracting the rules, we find the most frequently observed value in each element of condition part of the classifier in the action set generated in the external state, and calculate the percentage of such a specific value taken for each element of condition part.

Fig. 16 shows an example in which the first element takes the value of 0 in one out of five classifiers. Therefore, the ratio is calculated as 1/5 = 0.2. If the ratio in a certain element is high, the corresponding element can be regarded as an important element for deriving an optimal action.

Fig. 17 shows that the calculated ratio for each element related to "Cell 4" in woods1012 (see Fig. 18).
In this experiment, top 3 ratios are focused, and the corresponding 3 elements are chosen as the elements in which the values of 0 or 1 are determined. In both the original XCSMH and the EMOXCSMH, the extracted rule fires on the states corresponding to Cell 4. From the viewpoint of generality of the rules, the EMOXCSMH is superior to the original XCSMH. On the other hand, in the XCSMH with EORHG, the extracted rule fires on the two states corresponding to Cells 4 and 7. Considering the fact that the optimal policy for Cell 4 is different from that for Cell 7, the rule extracted from the XCSMH with EORHG is considered to be inferior from the viewpoint of availability of the extracted rules. The disadvantage of the XCSMH with EORHG is caused by the excessive degree of generality of the rules. Such a drawback is removed in the proposed EMOXCSMH, and we can conclude that the proposed EMOXCSMH has advantages in the sense not only of the optimality but also of the availability of the extracted rules.

5 Conclusion

In this paper, we have proposed an evolutionary multiobjective optimization-based classifier system in non-Markov environments, called EMOXCSMH, by taking account of not only the accuracy but also the generality of the rules obtained from the classifiers. First, we have defined the index function for measuring the degree of generality of the rules as the number of firing states. Next, we have proposed a revised NSGA-II so as to fit to the learning classifier systems by considering the ideas of active removal and elite preservation.

The experimental results have shown that the proposed EMOXCSMH can obtain a set of rules with higher degrees of generality than XCSMH, and that the EMOXCSMH can derive an optimal policy in non-Markov environments.

References


**Hideki Katagiri**
Hideki Katagiri is an Associate Professor at the Department of System Cybernetics, Graduate School of Engineering, Hiroshima University. His research and teaching activities are in the areas of operations research and soft computing, especially, fuzzy stochastic optimization, multiobjective decision making and data analysis.

**Ichiro Nishizaki**
Ichiro Nishizaki is a Professor at the Department of System Cybernetics, Graduate School of Engineering, Hiroshima University. His research and teaching activities are in the area of systems engineering, especially, game theory, multiobjective decision making and multi-agent systems.

**Tomohiro Hayashida**
Tomohiro Hayashida is an Assistant Professor at the Department of System Cybernetics, Graduate School of Engineering, Hiroshima University. His research and teaching activities are in the area of agent-based simulation analysis, game theory and decision-making theory.

**Keita Moriwake**
Keita Moriwake is a Master’s Degree Student at the Department of Artificial Complex Systems Engineering, Graduate School of Engineering, Hiroshima University. His current research interests are in the area of learning classifier systems.