Rule Extraction by Genetic Programming with Clustered Terminal Symbols

Akira Hara*[1], Haruko Tanaka*[2], Takumi Ichimura*[1], Tetsuyuki Takahama*[1]
1) Graduate School of Information Sciences, Hiroshima City University
3-4-1, Ozuka-higashi, Asaminami-ku, Hiroshima, 731-3194 Japan
email:{ahara, ichimura, takahama}@hiroshima-cu.ac.jp
2) Faculty of Information Sciences, Hiroshima City University

***Abstract*—When Genetic Programming (GP) is applied to rule extraction from databases, the attributes of the data are often used for the terminal symbols. However, in the case of the database with a large number of attributes, the search space becomes vast because the size of the terminal set increases. As a result, the search performance declines. For improving the search performance, we propose new methods for dealing with the large-scale terminal set. In the methods, the terminal symbols are clustered based on the similarities of the attributes. In the beginning of search, by reducing the number of terminal symbols, the rough and rapid search is performed. In the latter stage of search, by using the original attributes for terminal symbols, the local search is performed. By comparison with the conventional GP, the proposed methods showed the faster evolutional speed and extracted more accurate classification rules.**

I. Introduction

Genetic Programming (GP)[1] is an evolutionary optimization method of tree structural programs. The combination of functional and terminal symbols is changed by the genetic operations, and better solutions can be acquired. When GP is applied to rule extraction from a database, each attribute which forms the column of the database is often used for the terminal symbol[2], [3], [4]. In the case of the database with a large number of attributes, however, the search space becomes vast and the search performance declines. As one solution for this problem, several attributes which seem to be useful for rule expression may be selected beforehand. However, there is a possibility that the necessary attributes for the rule are not included in terminal symbols. This may cause the deterioration of accuracy of the rule.

In this research, we propose new rule extraction methods for the database with large-scale attributes. First, the groups of similar attributes are discovered by clustering methods. The resulting clusters are utilized for the setting of terminal symbols. In the beginning of search, rough and global search is performed by decreasing the number of terminals. In the latter stage of search, local search is performed by utilizing the information on the attributes within the same cluster.

II. Rule Extraction by GP

In this research, we focus on the rule extraction from the database by GP. Table I shows an example of database, where each instance has M attributes and a classification result (Class 0 or 1). IF_THEN rules can be acquired by using GP. An example of the IF_THEN rule is as follows:

TABLE I
Example of a database.

ID \ Attribute	A_1	A_2	A_3	$\ldots$	A_M	Class
1	0.2	0.5	0.2		0.7	0
2	0.2	0.2	0.3		0.4	1
3	0.1	0.5	0.6		0.3	0
4	0.9	0.3	0.5		0.5	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
N	0.5	0.9	0.2		0.3	1

IF ($A_1 > 0.3$) or (($A_3 < 0.4$) and ($A_M < 0.5$)) THEN Class 1

If the target class for rule extraction is decided, the consequent part of the rule is fixed. Therefore, the only antecedent part should be optimized by GP.

When the rule antecedent is expressed by the tree structural program, each attribute of data and the real values which can be used for thresholds of judgments are often used for terminal symbols T. That is, $T = \{A_1, A_2, \ldots, A_M, \Re\}$. In this case, the logical operators are used for functional symbols. That is, $F = \{and, or, >, <\}$. Fig.1 shows the tree structural representation of the rule in these settings.

The fitness is evaluated by the classification accuracy for the training data. By evolving the population, the more accurate classification rule can be acquired.

If we apply this method to the database with a large number of attributes, however, the size of terminal symbols becomes large. This causes an explosion of search space, and it results in the deterioration of search performance. Introducing data preparation process which decreases the number of attributes by the attribute selection may be efficient for search. However, the restriction of available attributes may cause the deterioration of the expressive power of the acquired rule.

III. GP with Clustered Terminal Symbols

In our approach, we do not select a subset of attributes relevant for the target data mining task. We utilize all the available attributes for rule extraction. The cluster structure of attributes is used for realizing the effective search. In

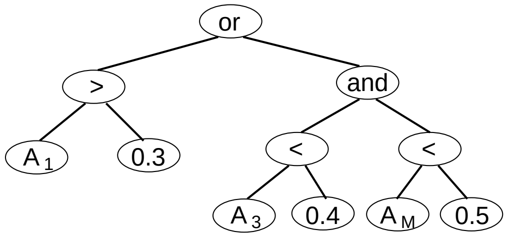

Fig. 1. Rule representation by tree structural program.

TABLE II
NORMALIZED DATABASE.

	A_1	...	A_i	...	A_M	class
Data 1	a_{11}	...	a_{1i}	...	a_{1M}	$class_1$
⋮	⋮		⋮		⋮	⋮
Data j	a_{j1}	...	a_{ji}	...	a_{jM}	$class_j$
⋮	⋮		⋮		⋮	⋮
Data N	a_{N1}	...	a_{Ni}	...	a_{NM}	$class_N$

our proposed method, the grouping of similar attributes are performed by the clustering methods. In the beginning of search, each cluster center is regarded as a new attribute, and the set of terminal symbols consists of the only cluster centers. This approach is aimed at reducing the search space. In the latter stage of search, the original attributes of database are used for improving the expressive ability of the rule. In addition, the improvement of search performance can be realized by the local search based on the cluster structure.

First, we describe how to apply the clustering method to the database attributes. The definition of the distance of the arbitrary two objects is necessary for clustering. We assume the database such as the Table I where N data is included and each instance has M attributes, $A_1, A_2, ..., A_M$. As the data preparation process, each column of the table is normalized. That is, the range of each column is converted to [0,1]. The equation of the normalization of data for the attribute $A_i (1 \leq i \leq M)$ is as follows:

$$a_{ji} = \frac{X_{ji} - \min_{A_i}}{\max_{A_i} - \min_{A_i}} \tag{1}$$

where X_{ji} represents the original value of the j-th data for attribute A_i, $\max_{A_i}$ represents the maximum value of the A_i in the original database, and $\min_{A_i}$ represents the minimum value of A_i. a_{ji} represents the normalized value of j-th data for attribute A_i ($0 \leq a_{ji} \leq 1$). Table II shows an example of the normalized database with N data and M attributes.

In usual machine learning settings, a row of a database may be regarded as a feature vector. In our proposed methods, however, we regard a column of the normalized database as the feature vector of the corresponding attribute. The clustering method is applied to the vectors for respective attributes. The distance between arbitrary attributes A_i and A_k, $d(A_i, A_k)$, is defined as follows:

TABLE III
DATABASE USING CLUSTER CENTERS.

-	k_1	...	k_K	class
1	k_{11}	...	k_{1K}	$class_1$
⋮	⋮		⋮	⋮
N	k_{N1}	...	k_{NK}	$class_N$

$$d(A_i, A_k) = \sqrt{\sum_{j=1}^{N} (a_{ji} - a_{jk})^2} \tag{2}$$

The clustering algorithms are either hierarchical or non-hierarchical such as K-means method. In this research, we propose two kinds of improved GP methods, GP using K-means clustering and GP using Hierarchical Clustering. In the next section, the details of respective methods are explained.

A. GP using K-means clustering (K-GP)

In the beginning of search by GP using K-means clustering, cluster centers by the K-means are treated as new attributes (terminals). The GP using K-means clustering method is called K-GP. The procedures of K-GP for the database with M attributes, $A_1, A_2, ..., A_M$, are as follows:

1) Normalization of database
   The values in each column of the database are normalized by equation (1).

2) Reduction of the number of attributes by K-means method
   The number of clusters K is set to be $K \leq M$. K-means clustering method is applied to the feature vectors for attributes.

3) GP using the cluster centers as the terminal symbols.
   It is regarded that the database is changed like Table III by the clustering step 2). In the Table III, k_{ji} represents the j-th value of cluster center k_i ($0 \leq \mathrm{k}_{ji} \leq 1, 1 \leq i \leq K, 1 \leq j \leq N$). This is the average value of data j on the attributes which belongs to the cluster k_i.
   The generation of initial individuals and genetic operators are performed like standard GP.

4) Replacement of terminal symbols
   At a pre-specified generation while executing step 3), all the attribute nodes, $k_i (1 \leq i \leq K)$, in individuals are replaced with A_x, which is $\min\{d(A_x, k_i)\}$ $(A_x \in k_i)$. The generation at which the replacement of terminals occurs is called the transition generation.

5) GP using original attributes
   The initial population of this step is the population to which terminal replacement has been applied. The original attributes, $A_1, A_2, ..., A_M$, are used for the terminals. When the mutation operation is applied to an attribute

node, a new node is selected randomly from the subset of terminals within the same cluster as the object attribute. For example, if the object attribute A_1 for mutation is the member of the cluster k_1, the newly selected attribute is necessarily included in the same cluster k_1. This local search technique will be useful for improving the search performance. The other genetic operations are performed like standard GP.

In addition, as variants of K-GP, K-GP$^-$ and c(combined)K-GP are also proposed. In K-GP$^-$, the step 3) is executed till the last generation without the subsequent replacement process. In cK-GP, the step 4) is not performed, and the step 5) is modified. The cluster center in the same cluster is also included in the newly selectable attributes for mutation. The terminal symbols after the transition generation consists of $(K + M)$ attributes (cluster center $k_1, ..., k_K$ and original attribute $A_1, ..., A_M$).

The cluster centers will appear in the acquired rules in K-GP$^-$ and cK-GP. When the acquired rule is applied to the coming data, the average of corresponding attributes is need to be calculated based on the cluster structure. For example, if the rule has a node k_2, which consists of A_1 and A_5, the average of A_1 and A_5 of the coming data becomes the value of k_2.

B. GP using hierarchical clustering (HC-GP)

The GP using Hierarchical Clustering method is called HC-GP. The procedures of HC-GP for the database with M attributes, $A_1, A_2, ..., A_M$, are as follows: (The description of the procedure is omitted if the process is the same as K-GP.)

1) Normalization of database

2) Reduction of the number of attributes by Hierarchical Clustering
The M attributes are clustered by the agglomerative hierarchical clustering method[5]. The single linkage method is used for the criteria for determining distance between arbitrary two clusters. The minimum number of clusters C is set to be $C \leq M$. The history of the clustering from the initial state, where each attribute represents its own cluster, to the termination state, where the number of clusters becomes C. That is, a treelike cluster structure with $(M-C)$ kinds of state is memorized. The C clusters at termination state are used at the initial generation.

3) Attribute Selection for GP terminals
The attributes used for GP terminals are decided in respective clusters by the following processes:
   a) For the current state of clustering, each cluster center is calculated.
   b) The nearest original attribute to the cluster center is selected for GP terminals.

   At the initial state, C attributes are selected for GP terminals. The attributes are represented by $c_1, c_2, ..., c_C (c_i \in \{A_1, ..., A_M\}, 1 \leq i \leq C)$. The selected attributes are called the representative attributes.

4) GP using the representative attributes
Initial population is generated by using the C representative attributes.
Though genetic operations are performed like standard GP, the system checks every generation whether the update of the best fitness value occurs. If the improvement of the fitness has not been occurred for G generations and the current number of clusters is less than M, the following process is performed.
   a) The clusters are divided based on the memorized cluster relationship till the clusters increases by the H clusters. The parameter H specifies the number of clusters which is added by a cluster division step. The range of H is $0 < H \leq (M - C)$.
   b) The representative attributes which have been selected before remain in the GP terminals. For the clusters which have no representative attributes in terminals, the new representative attributes are selected by the step 3).

   If the cluster division is performed t times from initial generation, the number of the representative attributes becomes $C + Ht$. Ht is changed from 0 to $(M - C)$.

IV. Experiments

A. Problem Settings

The MUSK clean1 database[6] is used for the experiments. In the database, each instance is classified into two classes, MUSK class or Non-MUSK class. The MUSK class data is regarded as the target class, and the rule which returns true for the target class data is extracted.

The properties of the database are as follows:

#Attributes: 166
Attributes Characteristics: Integer
Missing Attributes Values: None
#Instances: 476
Class Distribution: MUSKs:207, Non-MUSKs: 269

B. GP parameters

First, we define the symbols for representing attributes and clusters as follows: $A_1, ..., A_{166}$ represent the original attributes of MUSK database. $k_1, ..., k_K$ represent the K cluster centers acquired by the K-means method. $c_1, ..., c_C$ represent the C representative attributes based on the hierarchical clustering. By using the notations, we summarize briefly the five kinds of methods again.

- standard GP (sGP)
  sGP is used for the comparison with the proposed methods.
  All attributes are utilized for the terminal symbols. The set of terminal symbols is $T = \{A_1, ..., A_{166}, \Re\}$ $(0 \leq \Re \leq 1)$.

- K-GP$^-$
  K cluster centers by the K-means method are utilized for the terminal symbols. The set of terminal symbols is $T = \{k_1, ..., k_K, \Re\}$. Other settings are the same as sGP.
- K-GP
  First, K cluster centers by the K-means method are utilized for the terminal symbols. At the transition generation, each leaf node of individuals is replaced with the nearest-neighbor original attribute. After that, if the mutation is applied to the original attribute, a new attribute is selected from the members in the same cluster to which the original attribute belongs.
  The set of terminal symbols is as follows:
  Before the transition generation: $T = \{k_1, ..., k_K, \Re\}$
  After the transition generation: $T = \{A_1, ..., A_{166}, \Re\}$
- cK-GP
  First, K cluster centers by the K-means method are utilized for the terminal symbols. After the transition generation, if the mutation is applied to the original attribute or cluster center, a new attribute is selected from the members and the center of the same cluster to which the original attribute belongs.
  The set of terminal symbols is as follows:
  Before the transition generation: $T = \{k_1, ..., k_K, \Re\}$
  After the transition generation:
  $T = \{k_1, ..., k_K, A_1, ..., A_{166}, \Re\}$
- HC-GP
  The cluster structure is acquired by the hierarchical clustering. The representative attributes are selected in respective clusters and utilized for terminal symbols. The number of terminal symbols increases gradually by using the treelike cluster structure.
  The set of terminal symbols is as follows:
  Initial generation: $T = \{c_1, ..., c_C, \Re\}$
  Generation with t times division step:
  $T = \{c_1, ..., c_C, ..., c_{C'}, \Re\} (C' = C + Ht, C < C' \leq 166)$

The common parameter settings are as follows:

Functional Symbols: $F = \{and, or, <, >\}$
Population Size: 50
Max generation: 1000
Selection: Tournament Selection
Tournament Size: 2
#Elite individual: 1
Crossover rate: 0.8
Mutation rate: 0.2

For evaluating each individual, the classification results for instances in the database are categorized into four cases: true-positive (TP), true-negative (TN), false-positive (FP), and false-negative (FN)[2]. The classification accuracy as shown in equation (3) is used for the fitness function.

$$Fitness = \frac{TP + TN}{TP + FP + FN + TN} \quad (3)$$

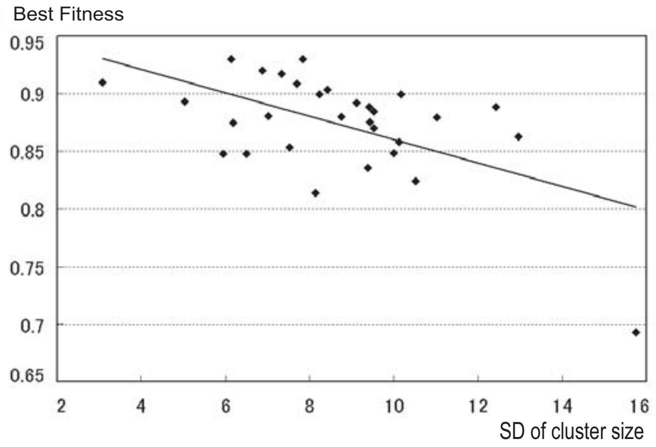

Fig. 2. Relationship between the best fitness and the standard deviation of cluster size.

C. Experimental Results

The following experimental results are averaged over 30 trials.

1) K-GP: K-GP utilize the result of clustering by K-means method. However, the set of cluster centers may be different due to the randomly selected initial cluster centers for K-means method. As preliminary experiments, the K-means clustering is performed for the MUSK database several times. In some cases, the number of members in respective clusters becomes almost uniform. In some cases, a part of clusters has the large number of members than other clusters. Therefore, we investigate the relationship between the bias of the number of members in the clusters and the best fitness under the terminal setting by using K-GP$^-$. The standard deviation of the number of members in the clusters is used for an indicator of the bias of cluster size. Respective plots in Fig.2 show the relationship between the acquired best fitness at the last generation and the standard deviation of the number of members in the clusters in each trial over 30 runs. The approximate line by the least square method is also drawn. For the MUSK database, the clusters with a low bias of the number of members tend to produce better fitness. In consideration for the results, the K-means method is performed 10 times, and the clustering with the minimum standard deviation on the number of members is utilized for the settings of terminal symbols in the following experiments.

Next, we investigate the influence of the parameter K on search performance by K-GP$^-$ method. Three kinds of K, 10, 20 and 30, are examined. The results are shown in Fig.3 and Table IV. The K-GP$^-$ shows better performance on any K values than sGP. The difference of the performance by the parameter K is not so apparent. However, the performance by $K = 10$ is lower than that by other settings after 80th generation. In contrast, the performance by $K = 30$ is lower till 80th generation. Therefore, if K (the number of terminals) is small, the evolutional speed is faster though the classification accuracy is lower. If K is large, the final

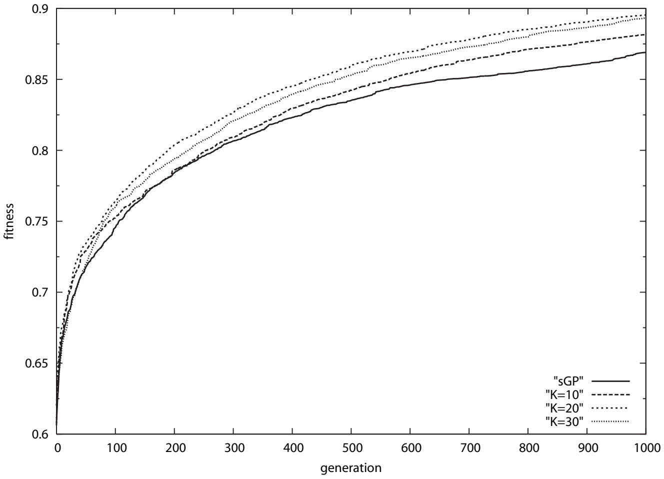

Fig. 3. The best fitness curves of K-GP^{-}.

TABLE IV
COMPARISON OF THE BEST FITNESS IN K-GP^{-}.

Methods	Best Fitness	Standard Deviation
sGP	0.868954	0.028182
K=10	0.881557	0.033841
K=20	0.895605	0.031388
K=30	0.893172	0.032556

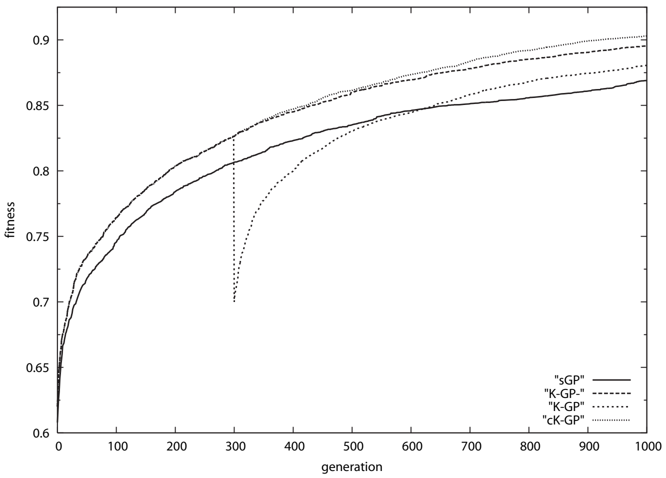

Fig. 4. The Best fitness curves of K-GP.

TABLE V
COMPARISON OF SGP AND K-GP.

Methods	Best Fitness	Standard Deviation
sGP	0.868954	0.028182
K-GP^{-}	0.895605	0.031388
K-GP	0.880573	0.029356
cK-GP	0.902939	0.029139

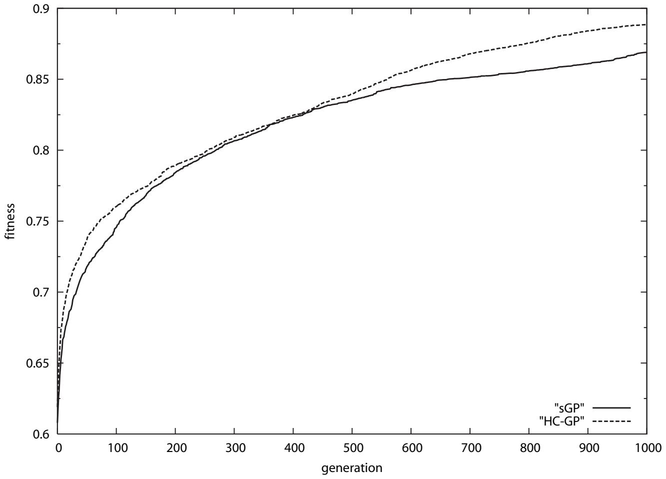

Fig. 5. The best fitness curves of HC-GP.

TABLE VI
COMPARISON OF SGP AND HC-GP.

Methods	Best Fitness	Standard Deviation
sGP	0.868954	0.028182
HC-GP	0.888653	0.027898

classification accuracy becomes higher though the evolutional speed is not so fast. In these experimental settings, $K = 20$ shows the best results by balancing evolutional speed with classification accuracy. Therefore, $K = 20$ is used for the following experiments. Considering how to set appropriate value of K depending on the database is one of the future works.

Next, we compared K-GP with cK-GP. The results are shown in Fig.4 and Table V. The transition generation in K-GP and cK-GP is set to 300th generation by preliminary experiments. As shown in Fig.4, there is a sudden drop in the best fitness at the transition generation in K-GP method. The cause of the sudden drop in the fitness is the difference between the cluster center and the nearest-neighbor original attribute. In cK-GP, the cluster centers remain in the rules without the replacement. Therefore, the drop in the fitness at the transition generation can be avoided. In addition, cK-GP has better performance than K-GP^{-}, because the original attributes can be also utilized.

2) HC-GP: In HC-GP, there are three parameters, the minimum number of clusters C, the period for judgment of cluster division G, and the increase volume of clusters at one separation step H. In order to avoid the decline in expressive ability of the rules, all the original attributes should be selected as representative ones up to the maximum generation. We perform experiments by using $C = 20, H = 20$ and $G = 15$ over 30 trials. In the case of these settings, all the original attributes were selected as representative attributes in 29 trials. The experimental results under these settings are shown in Fig.5 and Table VI. The HC-GP showed better performance than sGP.

3) Comparison among the proposed methods: We merged the results by respective proposed methods in Fig.6 and Table VII. Any proposed methods show better performance in classification accuracy than sGP. The difference of the

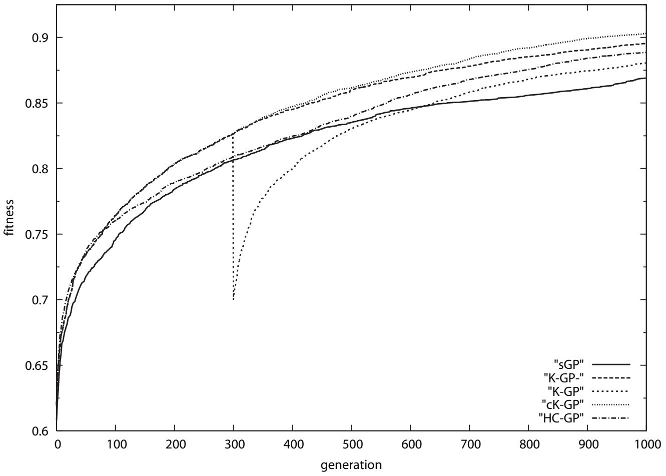

Fig. 6. The best fitness curves of the proposed methods.

TABLE VII
COMPARISON OF THE PROPOSED METHODS.

Methods	Best Fitness	Standard Deviation
sGP	0.868954	0.028182
K-GP^{-}	0.895605	0.031388
K-GP	0.880573	0.029356
cK-GP	0.902939	0.029139
HC-GP	0.888653	0.027898

best fitness between cK-GP and sGP is almost 0.034. This difference indicates that cK-GP can correctly classify about 16 instances more than sGP in the given database. In addition to the improvement of the classification accuracy, Fig.6 shows that the evolutional speed of the proposed methods becomes faster than sGP.

By comparison among proposed methods, K-GP^{-} and cK-GP are better than others for classification accuracy. Especially, cK-GP is better than K-GP^{-}, because cK-GP can use original attributes. However, the cluster centers may appear in the acquired rules in K-GP^{-} and cK-GP. Therefore, the comprehensibility of the acquired rules is not so high.

On the other hand, K-GP and HC-GP have an advantage that the acquired rule has high comprehensibility, because only the original attributes appear in the rule. In HC-GP, the genetic operations are the same as the sGP, though the terminal symbols increase gradually. There is scope for improvement of performance by modifying genetic operations in HC-GP.

V. DISCUSSION

From the viewpoints of evolutional speed and rule's practicability, the features of the proposed methods are summarized as follows:

- K-GP^{-}
  Evolutional speed is fast because the number of terminals is small. The acquired rules are not so practical because the original attributes are not used.
- K-GP
  Evolutional speed is slower than that of K-GP^{-}, because the fitness falls by the replacement of attributes. The acquired rules are practical because the only original attributes are used.
- cK-GP
  cK-GP shows the best performance in classification accuracy among the proposed methods, because the original attributes can be used without the loss by replacement. The acquired rules are not so practical because cluster centers may remain in the rules.
- HC-GP
  Evolutional speed is faster than sGP, because the size of terminal set increases gradually. The acquired rules are practical because the only original attributes are used.

Therefore, if the classification accuracy is of primary importance, cK-GP should be adopted. On the other hand, if the comprehensibility of rules is also important, HC-GP is a suitable method.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed the improved GP methods for database with a large number of attributes. In the beginning of search, the size of terminal set is reduced by the clustering of attributes. In the latter stage of search, the local search is performed based on the cluster structure. The two kinds of clustering, K-means and Hierarchical Clustering, are used, and four methods, K-GP, K-GP^{-}, cK-GP, and HC-GP, are proposed. All the methods showed better performance than standard GP. We also examined advantages and disadvantages of the proposed methods.

From now on, to examine the generality and effectiveness of the proposed methods, we have to apply the methods to other databases with a larger number of attributes. The total consumed time of a GP run including procedures of clustering terminal symbols should be also examined. In addition, there are many variations of clustering methods. For example, the single linkage method is adopted for HC-GP. However, if other definitions of inter-cluster distance such as the complete linkage or the group average method are used, the performance of HC-GP may change. Therefore, we also have to examine which method to adopt for clustering terminal symbols. Moreover, by considering these results, we will have to develop a new method with both high classification accuracy and comprehensibility.

REFERENCES

[1] J. R. Koza, "Genetic Programming: On the Programming of Computers by Means of Natural Selection", MIT Press (1992).
[2] A. A. Freitas, "Data Mining and Knowledge Discovery with Evolutionary Algorithms", Springer (2002).
[3] C. C. Bojarczuk, H. S. Lopes, and A. A. Freitas, "Genetic Programming for Knowledge Discovery in Chest Pain Diagnosis", IEEE Engineering in Medicine and Biology, Vol.19, No.4, pp.38-44 (2000)
[4] A. Hara, T. Ichimura, T. Takahama, and Y. Isomichi, "Extraction of Rules from Coronary Heart Disease Database Using Automatically Defined Groups", Proc. of The Eighth Conference on Knowledge-Based Intelligent Information and Engineering Systems, LNAI-3214, pp.1089-1096 (2004).
[5] D. T. Larose, "Discovering Knowledge in Data", WILEY (2005)
[6] UCI Machine Learning Repository, "http://archive.ics.uci.edu/ml".