An Attribute Reduction Algorithm by Switching Exhaustive and Heuristic Computation of Relative Reducts

Yasu Kudo
College of Information and Systems
Muroran Institute of Technology
Mizumoto 27-1, Muroran 050-8585, Japan
Email: kudo@csse.muroran-it.ac.jp

Tetsuya Murai
Graduate School of Information Science and Technology
Hokkaido University
Kita 14, Nishi 9, Kita-ku, Sapporo 060-0814, Japan
Email: murahiko@main.ist.hokudai.ac.jp

Abstract—We propose a heuristic algorithm to compute as many relative reducts as possible from a decision table with numerous condition attributes. The proposed algorithm is based on generating many reduced decision tables that preserve discernibility of objects in the given decision table. Moreover, the proposed algorithm switches exhaustive attribute reduction and heuristic attribute reduction by the number of condition attributes in decision tables. Experimental results indicate that the proposed algorithm can generate many relative reducts from datasets that are difficult to compute all relative reducts.

Keywords—attribute reduction; heuristic algorithm; reduced decision table; rough set

I. INTRODUCTION

Attribute reduction is one of the most important and hot research topics in rough set theory proposed by Pawlak [10], [11], and there have been various studies about attribute reduction. As an exhaustive attribute reduction algorithm, Skowron and Rauszer [13] proposed an algorithm to compute all relative reducts of the given decision table using the concept of the discernibility matrix. However, they also proved that the computational complexity of computing all relative reducts in the given decision table is NP-hard [13]. Thus, there have been many proposals of heuristic algorithms to compute some relative reducts instead of computing all relative reducts [1], [2], [3], [4], [5], [6], [7], [8], [14], [16], [17], [18], [19], [20]. Almost all of these heuristic algorithms generate small number of candidates (or just one candidate) of relative reducts of the given decision table. Thus, by just applying heuristic attribute reduction algorithms directly to the given data, it is impossible to analyze attributes that do not appear in the candidates of relative reduct, and from the data analysis viewpoint, various relative reducts should be generated as many as possible even though the number of condition attributes of the given decision table is numerous.

In this paper, we propose an attribute reduction algorithm to compute as many relative reducts as possible from a decision table with numerous condition attributes. Our algorithm is based on generating many reduced decision tables that preserve discernibility of objects in the given decision table. Moreover, our algorithm switches exhaustive attribute reduction and heuristic attribute reduction by the number of condition attributes in decision tables.

The rest of this paper is organized as follows. In Section II, we review Pawlak's rough set theory as the background of this paper. In Section III, we propose an attribute reduction algorithm to compute as many relative reducts as possible based on generating many reduced decision tables and switching exhaustive / heuristic attribute reduction. In order to evaluate the performance of the proposed algorithm, we apply the proposed algorithm to datasets in UCI repository [15] in Section IV and discuss the experimental results and issues of the proposed algorithm in Section V. Finally, Section VI concludes this paper. Note that this paper is a revised and extended version of the authors' manuscript [9].

II. ROUGH SET

In this section, we review the rough set theory, in particular, decision tables, relative reducts, and discernibility matrices. Note that the contents of this section are based on [8], [12].

A. Decision table and lower and upper approximations

Generally, data analysis subjects by rough sets are described by decision tables. Formally, a decision table is characterized by the following triple:

\[ DT = (U, C, d), \]

where \( U \) is a finite and nonempty set of objects, \( C \) is a finite and nonempty set of condition attributes, and \( d \) is a decision attribute such that \( d \not\in C \). Each attribute \( a \in C \cup \{d\} \) is a function \( a : U \rightarrow V \), where \( V \) is a set of values of attributes.

Indiscernibility relations based on subsets of attributes provide classifications of objects in decision tables. For any set of attributes \( A \subseteq C \cup \{d\} \), the indiscernibility relation \( R_A \) is the following binary relation on \( U \):

\[ R_A = \{(x, y) \mid a(x) = a(y), \forall a \in A\}. \]
If a pair \((x, y)\) is in \(R_A\), then two objects \(x\) and \(y\) are indiscernible with respect to all attributes in \(A\). It is well-known that any indiscernibility relation is an equivalence relation and equivalence classes by an equivalence relation consist of a partition on the domain of the equivalence relation. In particular, the indiscernibility relation \(R_d\) based on the decision attribute \(d\) provides a partition \(D = \{D_1, \cdots, D_k\}\), and each element \(D_i \in D\) is called a decision class.

Classifying objects with respect to condition attributes provides approximation of decision classes. Formally, for any set \(B \subseteq C\) of condition attributes and any decision class \(D_i \in D\), we let:

\[
B(D_i) = \{x \in U \mid [x]_B \subseteq D_i\},
\]

(3)

\[
\overline{B}(D_i) = \{x \in U \mid [x]_B \cap D_i \neq \emptyset\},
\]

(4)

where the set \([x]_B\) is the equivalence class of \(x\) by the indiscernibility relation \(R_B\). The set \(B(D_i)\) and the set \(\overline{B}(D_i)\) are called lower approximation and upper approximation of the decision class \(D_i\) with respect to \(B\), respectively. Note that the lower approximation \(B(D_i)\) is the set of objects that are correctly classified to the decision class \(D_i\) by checking all attributes in \(B\). A decision table is called consistent if and only if \(C(D_i) = D_i = \overline{C}(D_i)\) holds for all decision classes \(D_i \in D\).

**Example 1:** Table I is an example of a decision table we use in this paper, and consists of the following objects: \(U = \{x_1, \cdots, x_7\}\), \(C = \{c_1, \cdots, c_8\}\), and \(d\). The decision attribute \(d\) provides the following three decision classes: \(D_1 = \{x_1, x_5\}\), \(D_2 = \{x_3, x_4, x_7\}\) and \(D_3 = \{x_2, x_6\}\).

**B. Relative reduct**

By checking values of all condition attributes, we can classify all discernible objects of the given decision table to the corresponding decision classes. However, not all condition attributes may need to be checked in the sense that some condition attributes are essential to classify and the other attributes are redundant. A minimal set of condition attributes to classify all discernible objects to correct decision classes is called a relative reduct of the decision table.

For any subset \(X \subseteq C\) of condition attributes in a decision table \(DT\), we let:

\[
POS_X(D) = \bigcup_{D_i \in D} X(D_i).
\]

(5)

The set \(POS_X(D)\) is called the positive region of \(D\) by \(X\). All objects \(x \in POS_X(D)\) are classified to correct decision classes by checking all attributes in \(X\). In particular, the set \(POS_C(D)\) is the set of all discernible objects in \(DT\).

Here, we define relative reducts formally. A set \(A \subseteq C\) is called a relative reduct of the decision table \(DT\) if the set \(A\) satisfies the following conditions:

1) \(POS_A(D) = POS_C(D)\).
2) \(POS_B(D) \neq POS_C(D)\) for any proper subset \(B \subset A\).

Note that, in general, there are plural relative reducts in a decision table. The common part of all relative reducts is called the core of the decision table.

For example, there are the following four relative reducts in Table I: \(\{c_1, c_8\}\), \(\{c_4, c_8\}\), \(\{c_5, c_8\}\), and \(\{c_2, c_3, c_8\}\). The condition attribute \(c_8\) appears in all relative reducts in Table I and therefore the core of Table I is \(\{c_8\}\).

**C. Discernibility matrix**

The discernibility matrix is one of the most popular methods to compute all relative reducts in the decision table. Let \(DT\) be a decision table with \(|U|\) objects, where \(|U|\) is the cardinality of \(U\). The discernibility matrix \(DM\) of \(DT\) is a symmetric \(|U| \times |U|\) matrix whose element at \(i\)-th row and \(j\)-th column is the following set of condition attributes to discern between two objects \(x_i\) and \(x_j\):

\[
\delta_{ij} = \{a \in C \mid a(x_i) \neq a(x_j)\}.
\]

(6)

Each element \(a \in \delta_{ij}\) represents that \(x_i\) and \(x_j\) are discernible by checking the value of \(a\).

Using the discernibility matrix, we get all relative reducts of the decision table as follows:

1) Construct the following logical formula \(L(\delta_{ij})\) from each nonempty set \(\delta_{ij} = \{a_{k_1}, \cdots, a_{k_l}\}\) (\(i > j\) and \(l \geq 1\)) in the discernibility matrix:

\[
L(\delta_{ij}) : a_{k_1} \lor \cdots \lor a_{k_l}.
\]

(7)

2) Construct a conjunctive normal form \(\bigwedge_{i>j} L(\delta_{ij})\).
3) Transform the conjunctive normal form to the minimal disjunctive normal form:

\[
\bigwedge_{i>j} L(\delta_{ij}) \equiv \bigvee_{p=1}^{s} \bigwedge_{q=1}^{t_p} a_{pq}.
\]

(8)

4) For each conjunction \(a_{p_1} \lor \cdots \lor a_{p_t}\) (\(1 \leq p \leq s\)) in the minimal disjunctive normal form, construct a relative reduct \(\{a_{p_1}, \cdots, a_{p_t}\}\).

**Example 2:** Table II describes the discernibility matrix of the decision table by Table I. Each nonempty set that appears in the matrix represents the set of condition attributes that we should check to discern the corresponding objects. For example, the set \(\delta_{21} = \{c_1, c_3, c_4, c_8\}\) represents that we can distinguish between the objects \(x_2\) and \(x_1\) by comparing values of these objects of at least one of the condition attributes.
attributes $c_1$, $c_3$, $c_4$, and $c_5$ in $\delta_{21}$. Note that we omit
upper triangular components of the discernibility matrix in
Table II because the discernibility matrix is symmetric by
the definition. We construct a conjunctive normal form by
connecting logical formulas based on nonempty elements in
Table II by (7) and (8), and transform the conjunctive normal
form to the minimal disjunctive normal form as follows:
\[
(c_1 \lor c_3 \lor c_4 \lor c_5) \land (c_1 \lor c_2 \lor c_3 \lor c_5 \lor c_6 \lor c_8) \\
\quad \land \cdots \land (c_1 \lor c_2 \lor c_4 \lor c_5 \lor c_7 \lor c_8) \\
\equiv (c_1 \land c_8) \lor (c_4 \land c_8) \lor (c_5 \land c_8) \lor (c_2 \land c_3 \land c_8).
\]
Consequently, from this minimal disjunctive normal form,
we have the four relative reducts \{c_1, c_8\}, \{c_4, c_8\}, \{c_5, c_8\},
and \{c_2, c_3, c_8\}.

### III. Heuristic Algorithm for Attribute Reduction Using Reduced Decision Tables

In this section, we propose an algorithm to generate as
many relative reducts as possible from decision tables with
numerous condition attributes.

First, we define a concept of reduced decision tables of
the given decision table.

**Definition 1:** Let $DT = (U, C, d)$ be a decision table. A
reduced decision table of $DT$ is the following triple:

\[
RDT = (U, C', d),
\]
where $U$ and $d$ are identical to $DT$. The set of condition
attributes $C'$ satisfies the following conditions:

1) $C' \subseteq C$.

2) For any objects $x_i$ and $x_j$ that belong to different
decision classes, if $x_i$ and $x_j$ are discernible by $R_C$, $x_i$ and $x_j$ are also discernible by $R_{C'}$.

The reduced decision table preserves discernibility of
objects in the given decision table. In general, there are plural reduced decision tables of the given decision table.

Next, we introduce an algorithm to construct a reduced
decision table. It is easy to confirm that Algorithm 1 gen-
erates a reduced decision table of the given decision table.

In Algorithm 1, we select condition attributes from $C$ at
random based on the parameter of base size $b$, and supply
some attributes in elements of the discernibility matrix to
preserve discernibility of objects in the given decision table.

For generating as many relative reducts as possible from a
given decision table with numerous condition attributes, we
need to avoid generating the same reduced decision table as
long as possible when we use Algorithm 1 repeatedly. Thus,
randomness in selecting condition attributes is important to
keep variety of reduced decision tables.

### Algorithm 1: $dt$-reduction algorithm

**Input:** decision table $DT = (U, C, d)$,
discernibility matrix $DM$ of $DT$,
base size $b$

**Output:** reduced decision table $(U, C', d)$

1: Select $b$ attributes $a_1, \ldots, a_b$ from $C$ at random by
sampling without replacement
2: $C' = \{a_1, \ldots, a_b\}$
3: for all $\delta_{ij} \in DM$ such that $i > j$ do
4: if $\delta_{ij} \not= 0$ and $\delta_{ij} \cap C' = \emptyset$ then
5: Select $c \in \delta_{ij}$ at random
6: $C' = C' \cup \{c\}$
7: end if
8: end for
9: return $(U, C', d)$

Note that the base size $b$ decides the minimum number
of condition attributes of the reduced decision table and we
need to set $b$ appropriately. If $b$ is too large, it is obvious
that there is no merit of using Algorithm 1 at all because
the reduced decision table constructed by Algorithm 1 is
almost the same (or identical in the case of $b = |C|$) to the
original decision table. On the other hand, if $b$ is too small,
almost attributes in the reduced decision table are supplied at
steps 3-8 in Algorithm 1 to preserve discernibility of objects,
which is almost the same to generate a candidate of relative
reduct by random selection and it is not our intention.

The following theorem is essential for the attribute reduc-
tion algorithm that we propose later.

**Theorem 1:** Let $DT = (U, C, d)$ be a decision table and
$RDT = (U, C', d)$ be a reduced decision table of $DT$
constructed by Algorithm 1. A set of condition attributes $A \subseteq C'$ is a relative reduct of $RDT$ if and only if $A$ is a relative reduct of $DT$.

Proof: Because the set of objects $U$ and the decision attribute $d$ of $DT$ and $RDT$ are identical each other by Definition 1, it is obvious that decision classes of both $DT$ and $RDT$ are also identical. Thus, by the definition of relative reducts, it is sufficient to prove that $POS_{C'}(D) = POS_C(D)$ holds. First, we show $POS_{C'}(D) \subseteq POS_C(D)$ holds. Let $x \in POS_{C'}(D)$. Then, there exists a decision class $D_i$ such that $[x]_{C'} \subseteq D_i$ holds. The condition 1) in Definition 1 implies $R_{C'} \subseteq R_C$, and therefore $[x]_C \subseteq [x]_{C'}$ holds, which concludes $x \in POS_C(D)$. Conversely, we show $POS_C(D) \subseteq POS_{C'}(D)$ holds. Suppose that $x \notin POS_{C'}(D)$. Then, there exists an object $y$ such that $x$ and $y$ belong to different decision classes and $y \in [x]_{C'}$ holds, i.e., $(x, y) \in R_{C'}$. By the contraposition of the condition 2) in Definition 1, $(x, y) \in R_{C'}$ implies $(x, y) \in R_C$, and therefore, $y \in [x]_C$ holds. This implies $x \notin POS_C(D)$, which concludes the proof of the theorem.

As an example, we show how Algorithm 1 works to generate a reduced decision table from Table I.

Example 3: Using Algorithm 1, we construct a reduced decision table $RDT$ of the decision table $DT = (U, C, d)$ described by Table I. Let $DM$ be the discernibility matrix of $DT$ by Table II, and $b = 2$ be the base size as an input of Algorithm 1. At steps 1 and 2 of Algorithm 1, suppose two condition attributes $c_1$ and $c_4$ are selected with respect to $b = 2$, and let $C' = \{c_1, c_4\}$. At step 4, because $\delta_{T1} = \{c_8\} \neq \emptyset$ and $C' \cap \delta_{T1} = \emptyset$ hold, $C'$ is updated to $C' = \{c_1, c_4, c_8\}$ at step 5. Finally, Algorithm 1 generates a reduced decision table $RDT = (U, C', d)$ with $C' = \{c_1, c_4, c_8\}$ described by Table III. $RDT$ has the following two relative reducts: $\{c_1, c_8\}$ and $\{c_4, c_8\}$. As we described in Example 2, these are also the relative reducts of Table I.

Thus, we can generate relative reducts of the given decision table $DT$ by generating relative reducts of a reduced decision table $RDT$ constructed from $DT$ by Algorithm 1. If the number of condition attributes in $RDT$ is sufficiently small, we can generate all relative reducts of $RDT$ by exhaustive attribute reduction like the discernibility matrix. Thus, even though the number of condition attributes of the given decision table is numerous, generating reduced decision tables repeatedly and switching the methods of attribute reduction based on the size of each reduced decision table, we can generate many relative reducts (including candidates of relative reduct) instead of applying some heuristic attribute reduction directly to the original decision table.

Using Algorithm 1, we propose the following algorithm of attribute reduction based on generating reduced decision tables and switching exhaustive attribute reduction and heuristic attribute reduction.

**Algorithm 2** Exhaustive / heuristic attribute reduction

**Input:** decision table $DT = (U, C, d)$, base size $b$, size limit $L$, number of iteration $I$

**Output:** set of candidates of relative reduct $RED$

1: $RED = \emptyset$
2: $DM \leftarrow$ the discernibility matrix of $DT$
3: if $|C| \leq L$ then
4: \hspace{1em} $RED \leftarrow$ result of exhaustive attribute reduction from $DT$
5: \hspace{1em} \textbf{else}
6: \hspace{2em} for $i = 1$ to $I$ do
7: \hspace{3em} $RDT = \text{dtr}(DT, DM, b)$
8: \hspace{3em} if $|C'| \leq L$ then
9: \hspace{4em} \hspace{1em} $S \leftarrow$ result of exhaustive attribute reduction from $RDT$
10: \hspace{4em} \hspace{1em} $RED = RED \cup S$
11: \hspace{3em} \hspace{2em} end if
12: \hspace{2em} \hspace{1em} end if
13: \hspace{2em} end for
14: \hspace{1em} end if
16: \hspace{1em} return $RED$

Algorithm 2 switches exhaustive attribute reduction and heuristic attribute reduction by the size of decision tables. In Algorithm 2, the size limit $L$ is the threshold for switching attribute reduction methods and if the number of condition attributes of a decision table is smaller than $L$, Algorithm 2 tries to generate the set of all relative reducts of the decision table. Thus, we need to set the threshold $L$ appropriately. If the number of condition attributes of the given decision table $DT$ is greater than the threshold $L$, Algorithm 2 repeats $I$ times of generating a reduced decision table $RDT$ and attribute reduction from $RDT$ by selecting the exhaustive method or the heuristic method, and generate the set $RED$ of relative reducts. Note that $RED$ may contain some output with redundancy if the result of the heuristic attribute reduction is not guaranteed to generate relative reducts.

**IV. Experiments**

To evaluate the performance of the proposed method for attribute reduction using reduced decision tables, we applied the proposed heuristic algorithm to the following 10
datasets in the UCI machine learning repository [15]: Audiology, Annealing (Data), Annealing (Test), Lung Cancer, Cylinder Bands, Dermatology, Molecular Biology (Promoter Gene Sequences), Soybean (Small), Soybean (Large), and Sponge. In this experiment, we used the discernibility matrix as the exhaustive attribute reduction and a heuristic algorithm based on classification ability of condition attributes proposed by the authors [8] as the heuristic attribute reduction. We set the parameters used in Algorithm 2 as follows based on results of preliminary experiments: the base size $b = 1$, the size limit $L = 25$, and the number of iteration $I = 10$.

Table IV describes experiment results. In Table IV, the first column lists names of datasets. The second and third columns list the number of condition attributes and objects in each dataset. The fourth column represents the average number of condition attributes of reduced decision tables for each dataset. The fifth column represents the total number of outputs for each dataset generated by the proposed method. Finally, the sixth and seventh columns represent the average length of the output, i.e., the average number of attributes in outputs of the proposed algorithm, and the minimum length of outputs for each dataset, respectively.

For considering the influences of the base size, we also applied the proposed algorithm to the same 10 datasets in Table IV by the following parameters; the base size $b = 1$, the size limit $L = 25$, and the number of iteration $I = 10$, i.e., the base size was decreased from 10 to 1. Table V describes the results of comparison experiments. Similar to Table IV, Columns Tab., Red., Len., and Min. represent the average number of condition attributes of reduced decision tables, the total number of outputs, the average length of the output, and the minimum length of outputs for each dataset, respectively.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Audiology</td>
<td>69</td>
<td>200</td>
<td>23.6</td>
<td>25</td>
<td>15.68</td>
<td>13</td>
</tr>
<tr>
<td>Annealing-Data</td>
<td>38</td>
<td>798</td>
<td>15.8</td>
<td>24</td>
<td>8.46</td>
<td>7</td>
</tr>
<tr>
<td>Annealing-Test</td>
<td>38</td>
<td>100</td>
<td>11.7</td>
<td>19</td>
<td>5.21</td>
<td>4</td>
</tr>
<tr>
<td>Cylinder Bands</td>
<td>38</td>
<td>540</td>
<td>10.2</td>
<td>61</td>
<td>4.15</td>
<td>2</td>
</tr>
<tr>
<td>Dermatology</td>
<td>34</td>
<td>366</td>
<td>14.2</td>
<td>622</td>
<td>8.43</td>
<td>6</td>
</tr>
<tr>
<td>Lung Cancer</td>
<td>57</td>
<td>32</td>
<td>11.4</td>
<td>76</td>
<td>6.46</td>
<td>4</td>
</tr>
<tr>
<td>Promoter Gene</td>
<td>57</td>
<td>106</td>
<td>10</td>
<td>539</td>
<td>5.67</td>
<td>5</td>
</tr>
<tr>
<td>Soybean-Small</td>
<td>35</td>
<td>47</td>
<td>11.3</td>
<td>30</td>
<td>4.1</td>
<td>2</td>
</tr>
<tr>
<td>Soybean-Large</td>
<td>35</td>
<td>307</td>
<td>17.4</td>
<td>96</td>
<td>11.39</td>
<td>9</td>
</tr>
<tr>
<td>Sponge</td>
<td>45</td>
<td>76</td>
<td>13.9</td>
<td>78</td>
<td>7.83</td>
<td>5</td>
</tr>
</tbody>
</table>

Table V

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Audiology</td>
<td>17.3</td>
<td>19</td>
<td>13.95</td>
<td>13</td>
</tr>
<tr>
<td>Annealing-Data</td>
<td>10.7</td>
<td>20</td>
<td>7.3</td>
<td>7</td>
</tr>
<tr>
<td>Annealing-Test</td>
<td>7.2</td>
<td>15</td>
<td>4.87</td>
<td>4</td>
</tr>
<tr>
<td>Cylinder Bands</td>
<td>7</td>
<td>23</td>
<td>4.52</td>
<td>3</td>
</tr>
<tr>
<td>Dermatology</td>
<td>11</td>
<td>57</td>
<td>7.91</td>
<td>6</td>
</tr>
<tr>
<td>Lung Cancer</td>
<td>8</td>
<td>25</td>
<td>6.12</td>
<td>4</td>
</tr>
<tr>
<td>Promoter Gene</td>
<td>6</td>
<td>19</td>
<td>5.26</td>
<td>5</td>
</tr>
<tr>
<td>Soybean-Small</td>
<td>6.2</td>
<td>16</td>
<td>3.62</td>
<td>2</td>
</tr>
<tr>
<td>Soybean-Large</td>
<td>14.3</td>
<td>29</td>
<td>11.31</td>
<td>9</td>
</tr>
<tr>
<td>Sponge</td>
<td>11.9</td>
<td>39</td>
<td>7.67</td>
<td>6</td>
</tr>
</tbody>
</table>

V. DISCUSSION

As the experimental results in Table IV show, the proposed algorithm could generate many relative reducts for all 10 datasets. Except for two reduced decision tables of the Audiology dataset, exhaustive attribute reduction was used for attribute reduction from reduced decision tables for each dataset. The two cases of heuristic attribute reduction from the reduced decision tables also generated relative reducts, and therefore, all outputs in the experimental results had no redundant attributes.

In Table V, the average number of condition attributes of reduced decision tables and the total number of outputs were decreased rather than the results in Table IV for all datasets, even though exhaustive attribute reduction was used for all reduced decision tables of each dataset. Because the base size is $b = 1$, it is natural that the average numbers of condition attributes in reduced decision table are decreased, however, as discussed in Section III, it also causes loss of variety of generated relative reducts.

These experimental results in Table IV and Table V indicate that it is difficult to set appropriate values of the base size $b$, the size limit $L$, and the number of iteration $I$ because it depends on the dataset. For example, the base size $b = 10$ in Table IV may be too large for the Molecular Biology (Promoter Gene Sequences) dataset, because all the 10 reduced decision tables have no additionally supplied condition attributes in steps 3-8 in Algorithm 1. On the other hand, $b = 10$ may be small for the Audiology dataset, because more than 10 condition attributes are supplied to preserve the discernibility of the original decision table in average. These examples indicate that we need to adjust the base size $b$ by considering, for example, the number of objects, condition attributes, and decision classes of the given dataset. Moreover, in these experiments, we fixed the size limit $L = 25$ and the number of iteration $I = 10$ based on results of preliminary experiments, however, there is no basis of these settings of parameters. Thus, similar to the case of the base size $b$, we also need to consider adaptive setting of parameters $L$ and $I$. For example, it may be good for data analysis that all condition attributes appeared at least once in the output of the proposed algorithm. Thus, from the data analysis viewpoint, we may need to repeat generating a reduced decision table and attribute reduction until all condition attributes appear in the output. Moreover, instead of merely comparing the number of condition attributes in the given decision table with the threshold $L$, adaptive
Finally, because there is no dependency between reduced decision tables, we can compute attribute reduction of each reduced decision table separately and independently. Thus, we think that the proposed algorithm is quite useful for parallelization of parallelized attribute reduction system based on the proposed algorithm is an interesting future issue.

VI. CONCLUSION

In this paper, we proposed an attribute reduction algorithm to compute as many relative reducts as possible from a decision table with numerous condition attributes. Our algorithm is based on generating many reduced decision tables that preserve discernibility of objects in the given decision table and switching exhaustive attribute reduction and heuristic attribute reduction by the number of condition attributes in decision tables. The proposed algorithm was applied to 10 datasets in UCI machine learning repository, and generated many relative reducts of all datasets. These experimental results indicate that the proposed algorithm is applicable to datasets that are difficult to compute all relative reducts for generating as many relative reducts as possible. Future issues include adaptive setting of parameters relative to datasets, refinement of the switching method of exhaustive attribute reduction and heuristic attribute reduction, parallelization of attribute reduction using the proposed algorithm and an increased number of experiments using extensive datasets with more numerous condition attributes.

REFERENCES


