

Attribute reduction: a horizontal data decomposition approach

Piotr Hońko

Published online: 21 December 2014

© The Author(s) 2014. This article is published with open access at Springerlink.com

Abstract Ever-growing data generate a need for new solutions to the problem of attribute reduction. Such solutions are required to deal with limited memory capacity and with many computations needed for large data processing. This paper proposes new definitions of attribute reduction using horizontal data decomposition. Algorithms for computing reducts of an information system and decision table are developed and evaluated. In the proposed approach, the size of subtables obtained during the decomposition can be arbitrarily small. The reduct sets of subtables are computed independently from one another using any heuristic method for attribute reduction. Compared with standard attribute reduction methods, the proposed approach can produce the same reducts with less space complexity and with the same or less theoretical time complexity. Experiments conducted under this work show that for information systems with fewer attributes or reducts the time needed for computing the reduct set can be shorten.

Keywords Attribute reduction · Rough set theory · Data decomposition · FPGA

1 Introduction

Attribute reduction (also called feature selection) being a challenging task in areas such as data mining or pattern recognition has found many varied solutions. Rough set theory (Pawlak 1991) as a mathematical tool to deal with inconsis-

tent data has commonly been used to investigate this issue from a theoretical and practical viewpoint (e.g. Skowron and Rauszer 1992; Swiniarski 2001; Kryszkiewicz 2001).

A practical aspect for attribute reduction is clearly visible if the data to be processed are large. Removing unimportant attributes can significantly reduce the data size. However, due to memory capacity limitation, the data can be too large to process all at once.

This limitation caused the development of methods for decomposing data for attribute reduction (e.g. Deng and Huang 2006; Hu et al. 2006; Ye and Wu 2010). Data can be split with respect to objects (horizontal data decomposition) or to attributes (vertical data decomposition). Horizontal decomposition, in contrast to the vertical one, produces data subsets that are complete in terms of characteristics of objects (i.e. all attributes are stored in each subset). This makes it possible to process each subset separately and to use the same method as for the whole data to obtain partial results.

When applying data decomposition, the crucial problem is to divide the data set into subsets so that they are consistent with the original data. It means that no essential information is lost during the decomposition (e.g. Nguyen et al. 1998; Suraj 1996; Slezak 1999).

Another limitation related to ever-increasing data is the time needed to process large data sets. Data decomposition methods, however, are mainly intended to reduce the space complexity. A decrease in the time complexity is more difficult to achieve since additional operations are needed to merge the results obtained for subtables.

An efficient solution for time-consuming operations can be a hardware implementation of the method to be applied. This can be done using field programmable gate array (FPGA) devices. They are integrated circuits that can be programmed by the user after manufacturing. FPGA devices

Communicated by V. Loia.

P. Hońko (✉)
Faculty of Computer Science, Białystok University of Technology,
Wiejska 45A, 15-351 Białystok, Poland
e-mail: p.honko@pb.edu.pl

are used to implement software algorithms in hardware to increase their performance. They can significantly shorten the computation time due to parallelism, and also reduce power consumption and heat generation.

The main limitations of hardware implementation compared with the software one are the lack of flexibility in data processing and small memory capacity. Therefore, the method before its implementation can need a considerable adaptation to meet the requirements of the FPGA architecture (Grzes et al. 2013).

The goal of this paper is to develop an attribute reduction approach that can address the mentioned-above issues. This approach uses horizontal data decomposition for computing reducts of an information systems and decision table. Firstly, the universe is split into subsets (called middle subtables) according to a given data size. Then, any two subtables are joined forming the final subtables. For each such a subtable the set of reducts (called subreducts) is computed using any standard attribute reduction method. Finally, all subreduct sets are joined into one set that after reduction corresponds to the reduct set computed for the whole data. The space complexity of this approach is determined by the size of final subtables. The time complexity equals to or is less than that of the discernibility matrix-based attribute reduction method (Skowron and Rauszer 1992) applied to the whole table. The size of subtables can be arbitrary small (at least two objects) and subreducts are computed independently from one another, thanks to this the approach is suitable for hardware implementation using an FPGA.

The proposed approach overcomes the problem of memory capacity limitation since the final reduct set can be computed from the data that are split into suitable small portions. As the experiments show, for some datasets the computation time can be somewhat shortened. Based on the preliminary tests conducted in this paper and in Grzes et al. (2013) one can conclude that hardware implementation of the approach can considerably accelerate the attribute reduction process.

The following sections review existing attribute reduction methods (Sect. 2), restate basic notions related to attribute reduction in rough set theory (Sect. 3), propose new definitions that use horizontal data decomposition for computing reducts (Sect. 4), develop and evaluate algorithms for attribute reduction (Sect. 5), conduct experiments (Sect. 6) and discuss the approach (Sect. 7), review related works (Sect. 8), and provide concluding remarks (Sect. 9). The paper ends with an appendix that includes proof of propositions and theorems introduced in the paper.

2 Attribute reduction methods

In recent decades, many methods for finding reducts have been proposed in rough set theory. Most of them can be

categorized into the following general groups: discernibility matrix-based methods (Skowron and Rauszer 1992), positive region-based methods (Pawlak 1991; Grzymala-Busse 1991) and information entropy-based methods (Ślęzak 2002). Some of these methods are able to compute all reducts, other are developed to find one reduct, in particular, a minimal reduct.

Because of the scope of the paper, this section focuses on methods for computing all reducts. Most of them can be categorized into two groups: traversing based strategy and discernibility matrix-based strategy. The former relies on searching the whole powerset of the attribute set and checking subsets for being reducts. The latter computes for each pair of objects a subset of attributes that are needed to discern the objects. Reducts are constructed based on the attribute subsets that are minimal in the sense of inclusion. Both the strategies produce the same reducts but the discernibility matrix-based approach is less time consuming.

Skowron and Rauszer (1992) introduced a discernibility matrix and its alternative representation in the form of discernibility function to find all reducts in an information system. The idea of discernibility matrix/function has been intensively studied by many researchers. Hu and Cercone (1995) proposed a modification of the discernibility matrix for attribute reduction in a decision table. Kryszkiewicz (1998) applied discernibility functions in incomplete information systems and decision tables. Ye and Chen (2002) presented a discernibility matrix for attribute reduction in consistent and inconsistent decision tables. Zhang et al. (2003) studied finding reducts of inconsistent information systems. Degang et al. (2007) used the discernibility matrix for finding reducts in consistent and inconsistent covering decision system. The discernibility matrix/function has also become a starting point for computing all reducts by adapting different well-known strategies, e.g. exhaustive and genetic search (Bazan et al. 2000).

The main problem to face when developing a method for finding reducts is the computational complexity of the attribute reduction task. Finding all reducts is proven to be an NP-hard problem (Skowron and Rauszer 1992). Much effort has therefore been made to accelerate the attribute reduction process.

Susmaga (1998) constructed an absorbed discernibility list based on the discernibility matrix. The list includes minimal elements of the matrix that are sorted in ascending order according to their cardinalities. Reducts are computed based on the list using the breadth-first search method. Starzyk et al. (1999) sped up computation of all reducts based on the discernibility function thanks to the application of the absorption and expansion laws as well as the concept of strong equivalence. Two attributes that are locally strongly equivalent in the discernibility function are replaced with a single attribute. Leung and Li (2003) used maximal consistent

blocks as units to construct a reduced discernibility matrix of an incomplete information system. A maximal consistent block is understood as the maximal collection of objects in which all objects are indiscernible. To decrease the computational complexity of the classical rough set-based method, Li and Zhu (2005) defined an indiscernibility matrix based on the discernibility one. The computations in that approach are simplified because elements in the indiscernibility matrix are ordered. Yang (2006) proposed modifications of the discernibility matrix by dividing the universe into the consistent and inconsistent parts. That approach significantly reduces the time needed for computing reducts. Tsang et al. (2008) generalized the discernibility matrix method and applied it for covering generalized rough sets. To reduce the storage space used by the discernibility matrix methods, Xu et al. (2009) proposed a novel data structure, i.e. an improved frequent pattern tree. Unnecessary elements to compute reducts are not stored in the discernibility matrix. Wang and Ma (2009) developed a feature forest based algorithm for finding all reducts in consistent decision tables. To represent the data, the algorithm employs a decision forest that decreases the storage space. Reducts are computed based on the discernibility function in disjunctive normal form. Chen et al. (2012) proposed sample pair selection to construct the simplest form of the discernibility function of a decision table. Thanks to that, only minimal elements of the discernibility matrix are used to compute reducts. Zhang et al. (2013) developed a combinatorial optimization algorithm for finding all reducts of a covering decision system. The proposed attribute reduction is oriented towards obtaining the compact rules that are useful in making decision. Thi and Giang (2013) used Sperner system and the concept of minimal sets of an attribute to find all reducts of a consistent decision table.

Much research has also been devoted to finding a reduct, especially a minimal one. Although one reduct is sufficient to reduce the attribute set, the problem of finding all reducts still has its justification. A deeper analysis of the data can be conducted when all reducts are known. For instance, all reducts are necessary to find stable reducts (Bazan et al. 1994).

As shown in this section, the problem of improving the attribute reduction process by reducing the storage space or time-consuming calculations has been widely investigated. Another direction for making attribute reduction methods more efficient for large datasets is to divide the attribute reduction problem into subproblems. It can be done by applying data decomposition-based attribute reduction approaches.

3 Basic notions

This section restates basic definitions from rough set theory related to attribute reduction.

To store data to be processed, an information system is used.

Definition 1 (Pawlak 1991) (*Information system*) An information system is a pair $IS = (U, A)$, where U is a non-empty finite set of objects, called the universe, and A is a non-empty finite set of attributes.

Each attribute $a \in A$ is treated as a function $a : U \rightarrow V_a$, where V_a is the value set of a .

Essential information about data is expressed by an indiscernibility relation.

Definition 2 (Pawlak 1991) (*Indiscernibility relation*) An indiscernibility relation $IND(B)$ generated by $B \subseteq A$ on U is defined by

$$IND(B) = \{(x, y) \in U \times U : \forall a \in B a(x) = a(y)\} \quad (1)$$

The relation is used to define a reduct of the attribute set.

Definition 3 (Pawlak 1991) (*Reduct of attribute set*) A subset $B \subseteq A$ is a reduct of A on U if and only if

1. $IND(B) = IND(A)$,
2. $\forall C \subset B, C \neq \emptyset, IND(C) \neq IND(B)$.

The set of all reducts of A on U is denoted by $RED(A)$.

The reduct set of an information system can be computed using a discernibility function.

Definition 4 (Skowron and Rauszer 1992) (*Discernibility function*) A discernibility function f_{IS} of an information system $IS = (U, A)$ is a Boolean function of k Boolean variables a_1^*, \dots, a_k^* that correspond, respectively, to attributes $a_1, \dots, a_k \in A$ and is defined by

$$f_{IS}(a_1^*, \dots, a_k^*) = \bigwedge_{c_{x,y} \neq \emptyset} \bigvee_{a \in c_{x,y}} a^* \quad (2)$$

where $(c_{x,y})$ is the discernibility matrix of IS such that

$$\forall x, y \in U c_{x,y} = \{a \in A : a(x) \neq a(y)\}.$$

A prime implicant $a_{i_1}^* \wedge \dots \wedge a_{i_k}^*$ of f_{IS} is equivalent to a reduct $\{a_{i_1}, \dots, a_{i_k}\}$ of IS .

A special case of an information system, called decision table, is used to store data with class distribution.

Definition 5 (Pawlak 1991) (*Decision table*) A decision table is a pair $DT = (U, A \cup \{d\})$, where U is a non-empty finite set of objects, called the universe, A is a non-empty finite set of condition attributes, and $d \notin A$ is the decision attribute.

Each attribute $a \in A \cup \{d\}$ is treated as a function $a : U \rightarrow V_a$, where V_a is the value set of a .

For a decision table a relative indiscernibility relation and relative reduct of the attribute set are defined.

Definition 6 (Pawlak 1991; Miao et al. 2009) (*Relative indiscernibility relation*) A relative indiscernibility relation $IND(B, d)$ generated by $B \subseteq A$ on U is defined by

$$IND(B, d) = \{(x, y) \in U \times U : (x, y) \in IND(B) \vee d(x) = d(y)\} \tag{3}$$

Definition 7 (Pawlak 1991; Miao et al. 2009) (*Relative reduct of attribute set*) A subset $B \subseteq A$ is a relative reduct of A if and only if

1. $IND(B, d) = IND(A, d)$,
2. $\forall C \subseteq B, C \neq \emptyset, IND(C, d) \neq IND(B, d)$.

$U \setminus A$	Temperature	Headache	Weakness	Nausea	Flu
1	Very high	Yes	Yes	No	Yes
2	Normal	No	No	No	No
3	High	No	No	No	No
4	Normal	No	Yes	No	Yes
5	Normal	No	Yes	No	No
6	High	Yes	No	Yes	Yes
7	Very high	No	No	No	No
8	Normal	Yes	Yes	Yes	Yes

Treat the data table as the information system $IS = (U, A)$, where $U = \{1, \dots, 8\}$ and $A = \{\text{temperature, headache, weakness, nausea, flu}\}$. We obtain the following discernibility matrix $(c_{x,y})$ (For simplicity's sake the attributes names are abbreviated to their first letters and only the part of the matrix under the diagonal is shown since the matrix is symmetric.).

$$\left(\begin{array}{cccccccc} \emptyset & \{t, h, w, f\} & \{t, h, w, f\} & \{t, h\} & \{t, h, f\} & \{t, w, n\} & \{h, w, f\} & \{t, n\} \\ \emptyset & \{t\} & \{w, f\} & \{w\} & \{t, h, n, f\} & \{t\} & \{h, w, n, f\} & \{h, w, n, f\} \\ & \emptyset & \{t, w, f\} & \{t, w\} & \{h, n, f\} & \{t\} & A & A \\ & & \emptyset & \{f\} & \{t, h, w, n\} & \{t, w, f\} & \{h, n\} & \{h, n\} \\ & & & \emptyset & A & \{t, w\} & \{h, n, f\} & \{h, n, f\} \\ & & & & \emptyset & \{t, h, n, f\} & \{t, w\} & \{t, w\} \\ & & & & & \emptyset & A & A \\ & & & & & & \emptyset & \emptyset \end{array} \right)$$

The set of all relative reducts of A on U is denoted by $RED(A, d)$.

The relative reduct set of a decision table can be computed using a relative discernibility function.

Definition 8 (Skowron and Rauszer 1992) (*Relative discernibility function*) A relative discernibility function f_{DT} of a decision table $DT = (U, A \cup \{d\})$ is a Boolean function of k Boolean variables a_1^*, \dots, a_k^* that correspond, respectively, to attributes $a_1, \dots, a_k \in A$ and is defined by

$$f_{DT}(a_1^*, \dots, a_k^*) = \bigwedge_{c_{x,y}^d \neq \emptyset} \bigvee_{a \in c_{x,y}^d} a^* \tag{4}$$

where $(c_{x,y}^d)$ is the relative discernibility matrix of DT such that

$$\forall x, y \in U, c_{x,y}^d = \{a \in A : a(x) \neq a(y), d(x) \neq d(y)\}.$$

A prime implicant $a_{i_1}^* \wedge \dots \wedge a_{i_k}^*$ of f_{DT} is equivalent to a relative reduct $\{a_{i_1}, \dots, a_{i_k}\}$ of DT .

To illustrate the basic notions and those proposed in this paper, the following running example is used.

Example 1 Given a data table of patients who are suspected to be sick with flu.

The discernibility function derived from the matrix is $f_{IS}(t^*, h^*, w^*, n^*, f^*) = t^* \wedge w^* \wedge f^* \wedge (h^* \vee n^*)$. Based on its prime implicants $t^* \wedge h^* \wedge w^* \wedge f^*$ and $t^* \wedge w^* \wedge n^* \wedge f^*$ we obtain that the sets of reducts of IS are $RED(A) = \{\{t, h, w, f\}, \{t, w, n, f\}\}$.

The above data table can also be considered as the decision table $DT = (U, A' \cup \{flu\})$, where $A' = \{\text{temperature, headache, weakness, nausea}\}$.

We obtain in an analogous way that the set of relative reducts of DT is $RED(A', flu) = \{\{h, w\}, \{t, w, n\}\}$.

4 Horizontal decomposition for attribute reduction

This section proposes two attribute reduction approaches based on horizontal decomposition. The first one is developed for an information system, and the second one is for a decision table.

4.1 Horizontal decomposition of information system

Firstly, an indiscernibility relation on a subset of the universe is defined.

Definition 9 (*Indiscernibility relation on universe subset*)

An indiscernibility relation $IND(B)$ generated by $B \subseteq A$ on $X \subseteq U$ is defined by

$$IND_X(B) = IND(B) \cap X \times X \tag{5}$$

The relation has the following properties.

Proposition 1 *For any information system $IS = (U, A)$, any non-empty $B \subseteq A$ and $X \subseteq U$ the following hold.*

- 1.1 $IND(B) = IND_X(B)$ for $X = U$.
- 1.2 $IND_X(B) \subseteq IND(B)$.
- 1.3 $\forall_{x,y \in X} ((x, y) \in IND_X(B) \Leftrightarrow (x, y) \in IND(B))$.
- 1.4 $IND_X(B) = IND_X(A) \Rightarrow \exists_{\substack{C \subseteq A \\ B \subseteq C}} IND(C) = IND(A)$.
- 1.5 $IND(B) = IND(A) \Rightarrow \exists_{\substack{C \subseteq B \\ C \neq \emptyset}} IND_X(C) = IND_X(A)$.

A reduct of the attribute set on a subset of the universe is defined as follows.

Definition 10 (*Reduct of attribute set on universe subset*) A subset $B \subseteq A$ is a reduct of A on $X \subseteq U$ if and only if

- 1. $IND_X(B) = IND_X(A)$,
- 2. $\forall_{\substack{C \subseteq B \\ C \neq \emptyset}} IND_X(C) \neq IND_X(B)$.

The set of all reducts of A on $X \subseteq U$ is denoted by $RED_X(A)$.

To differ a reduct on a universe subset from one on the whole universe, the former is called subreduct.

To decompose data, a covering of the universe is constructed. Based on each set of the covering a middle subtable is formed. Each pair of middle subtables is merged into one final subtable. To compute reduct sets of an information system, the subreduct sets of all the final subtables are joined using the following operation.

Definition 11 (*Operation $\dot{\cup}$*) An operation $\dot{\cup}$ on families of sets is defined as follows

- 1. $\mathcal{S} \dot{\cup} \emptyset = \emptyset \dot{\cup} \mathcal{S} = \mathcal{S}$;
- 2. $\mathcal{S} \dot{\cup} \mathcal{S}' = \{S \cup S' : S \in \mathcal{S}, S' \in \mathcal{S}'\}$;
- 3. $\dot{\bigcup}_{i=1}^k \mathcal{S}_i = \mathcal{S}_1 \dot{\cup} \mathcal{S}_2 \dot{\cup} \dots \dot{\cup} \mathcal{S}_k$, where $k > 1$.

The family of attribute subsets created by the above operation includes, in general, not only reducts but also supersets of them. To remove unnecessary sets, the following operation is used. Let $\min(\mathcal{S})$ be the set of minimal elements of a family \mathcal{S} of sets partially ordered by the relation \subseteq .

The above operations have the following properties.

Proposition 2 *For any families $\mathcal{S}, \mathcal{S}'$ of sets the following hold.*

2.1 $\mathcal{S} \subseteq \mathcal{S} \dot{\cup} \mathcal{S}$.

2.2 $\min(\mathcal{S} \dot{\cup} \mathcal{S}) = \min(\mathcal{S})$.

2.3 $\min(\mathcal{S} \dot{\cup} \mathcal{S}') = \min(\min(\mathcal{S}) \dot{\cup} \min(\mathcal{S}'))$.

2.4 $\dot{\bigcup}_{\mathcal{S} \in \mathbf{S}} \mathcal{S} = \dot{\bigcup}_{\mathcal{S} \in \mathbf{S}} \mathcal{S} \dot{\cup} \dot{\bigcup}_{\mathcal{S}' \in \mathbf{S}'} \mathcal{S}'$, where \mathbf{S}, \mathbf{S}' are any families of families of sets.

These operations are used to define attribute reduction of an information system.

Theorem 1 *Let $IS = (U, A)$ be an information system, $\{X_1, X_2, \dots, X_k\}$ ($k > 1$) be a covering of U . The following holds*

$$RED(A) = \min \left(\dot{\bigcup}_{1 \leq i < j \leq k} RED_{X_i \cup X_j}(A) \right) \tag{6}$$

The above theorem is true for any total covering of the universe. From the practical viewpoint, its special case, i.e. a partition is taken.

Example 2 Consider a covering $\{X_1, X_2, X_3, X_4\}$ of U , where $X_1 = \{3, 8\}$, $X_2 = \{1, 7\}$, $X_3 = \{4, 5\}$, $X_4 = \{2, 6\}$.

We have the following subreducts $RED_{X_1 \cup X_2}(A) = \{\{t, h\}, \{t, w\}, \{t, f\}\}$, $RED_{X_1 \cup X_3}(A) = \{\{t, h, f\}, \{t, n, f\}, \{h, w, f\}, \{w, n, f\}\}$, $RED_{X_1 \cup X_4}(A) = \{\{t, h\}, \{t, n\}, \{t, f\}\}$, $RED_{X_2 \cup X_3}(A) = \{\{t, f\}, \{h, w, f\}\}$, $RED_{X_2 \cup X_4}(A) = \{\{t, h\}, \{t, w\}, \{t, f\}\}$, $RED_{X_3 \cup X_4}(A) = \{\{w, f\}\}$.

We obtain $\min(\dot{\bigcup}_{1 \leq i < j \leq 4} RED_{X_i \cup X_j}(A)) = \min(\min(RED_{X_1 \cup X_2}(A) \dot{\cup} RED_{X_1 \cup X_3}(A)) \dot{\cup} \min(RED_{X_1 \cup X_4}(A) \dot{\cup} RED_{X_2 \cup X_3}(A)) \dot{\cup} \min(RED_{X_2 \cup X_4}(A) \dot{\cup} RED_{X_3 \cup X_4}(A))) = \min(\{\{t, h, f\}, \{t, n, f\}\} \dot{\cup} \{\{t, f\}\} \dot{\cup} \{\{t, w, f\}\}) = \{\{t, h, w, f\}, \{t, n, w, f\}\} = RED(A)$.

Besides redundant operations that follow from using a covering of the universe instead of a partition of it, some other operations can be repeated unnecessarily when using the above definition. Namely, when we compute reducts on sets $X \cup Y$ and $X \cup Z$, we process each pair of objects from X twice. To exclude these repetitions, attribute reduction is defined in the following way.

Theorem 2 *Let $IS = (U, A)$ be an information system, $\{X_1, X_2, \dots, X_k\}$ ($k > 1$) be a covering of U . The following holds*

$$RED(A) = \min \left(\dot{\bigcup}_{1 \leq i \leq k} RED_{X_i}(A) \dot{\cup} \dot{\bigcup}_{1 \leq i < j \leq k} RED_{X_i \cup X_j}(A, d) \right) \tag{7}$$

where d is a virtual decision such that $\forall_{x,y \in U} (d(x) = d(y) \Leftrightarrow \exists_{1 \leq i \leq k} x, y \in X_i)$.

The use of attribute reduction on the set $X \cup Y$ with respect to the virtual decision (see Definition 13) guarantees that any object from X is processed with objects from Y only, and vice versa.

Example 3 Consider the covering of U from Example 2.

We have the following subreducts $RED_{X_1}(A) = \{\{t\}, \{h\}, \{w\}, \{n\}, \{f\}\}$, $RED_{X_2}(A) = \{\{h\}, \{w\}, \{f\}\}$, $RED_{X_3}(A) = \{\{f\}\}$, $RED_{X_4}(A) = \{\{t\}, \{h\}, \{n\}, \{f\}\}$, $RED_{X_1 \cup X_2}(A, d) = \{\{t\}\}$, $RED_{X_1 \cup X_3}(A, d) = \{\{t, h\}, \{t, n\}, \{h, w\}, \{w, n\}\}$, $RED_{X_1 \cup X_4}(A, d) = \{\{t, h\}, \{t, n\}, \{t, f\}\}$, $RED_{X_2 \cup X_3}(A, d) = \{\{t\}, \{h, w\}\}$, $RED_{X_2 \cup X_4}(A, d) = \{\{t\}\}$, $RED_{X_3 \cup X_4}(A, d) = \{\{w\}\}$.

We obtain $\min(\dot{\bigcup}_{1 \leq i \leq 4} RED_{X_i}(A) \dot{\cup} \dot{\bigcup}_{1 \leq i < j \leq 4} RED_{X_i \cup X_j}(A, d)) = \min(\min(\dot{\bigcup}_{1 \leq i \leq 4} RED_{X_i}(A)) \dot{\cup} \min(\dot{\bigcup}_{1 \leq i < j \leq 4} RED_{X_i \cup X_j}(A, d))) = \min(\{\{f\}\} \dot{\cup} \{\{t, h, w\}, \{t, n, w\}\}) = \{\{t, h, w, f\}, \{t, w, n, f\}\} = RED(A)$.

4.2 Horizontal decomposition of decision table

Firstly, a relative indiscernibility relation on a subset of the universe is defined.

Definition 12 (*Relative indiscernibility relation on universe subset*) A relative indiscernibility relation $IND(B)$ generated by $B \subseteq A$ on $X \subseteq U$ is defined by

$$IND_X(B, d) = IND(B, d) \cap X \times X \tag{8}$$

The relation properties are the following.

Proposition 3 For any decision table $DT = (U, A \cup \{d\})$, non-empty $B \subseteq A$ and $X \subseteq U$ the following hold.

- 3.1 $IND(B, d) = IND_X(B, d)$ for $X = U$.
- 3.2 $IND_X(B, d) \subseteq IND(B, d)$.
- 3.3 $\forall x, y \in X ((x, y) \in IND_X(B, d) \Leftrightarrow (x, y) \in IND(B, d))$.
- 3.4 $IND_X(B, d) = IND_X(A, d) \Rightarrow \exists_{\substack{C \subseteq A \\ B \subseteq C}} IND(C, d) = IND(A, d)$.
- 3.5 $IND(B, d) = IND(A, d) \Rightarrow \exists_{\substack{C \subseteq B \\ C \neq \emptyset}} IND_X(C, d) = IND_X(A, d)$.

A relative reduct of attribute set on a universe subset is defined as follows.

Definition 13 (*Relative reduct of attribute set on universe subset*) A subset $B \subseteq A$ is a relative reduct of A on $X \subseteq U$ if and only if

- 1. $IND_X(B, d) = IND_X(A, d)$,
- 2. $\forall_{\substack{C \subseteq B \\ C \neq \emptyset}} IND_X(C, d) \neq IND_X(B, d)$.

The set of all relative reducts of A on $X \subseteq U$ is denoted by $RED_X(A, d)$.

To decompose a decision table, each decision class is divided into subsets (middle subtables), then each pair of subsets of different classes is merged into one set (final subtables). To compute relative reduct sets of a decision table,

the subreduct sets of all the final subtables are joined (analogously to the information system case).

Theorem 3 Let $DT = (U, A \cup \{d\})$ be a decision table such that $U = \bigcup_{i=1}^k X_{v_i}$, where X_{v_i} is a decision class, $v_i \in V_d$ and $k > 1$ is the number of different classes in DT . Let \mathcal{X}_{v_i} be a covering of X_{v_i} ($1 \leq i \leq k$). The following holds

$$RED(A, d) = \min \left(\dot{\bigcup}_{\substack{1 \leq i < j \leq k \\ X \in \mathcal{X}_{v_i}, Y \in \mathcal{X}_{v_j}}} RED_{X \cup Y}(A, d) \right) \tag{9}$$

Example 4 Consider coverings $\{X_1, X_2\}$ and $\{X_3, X_4\}$ of, respectively, decision classes “no” and “yes”, where $X_1 = \{2, 7\}$, $X_2 = \{3, 5\}$, $X_3 = \{1, 4\}$, $X_4 = \{6, 8\}$.

We have the following relative subreducts $RED_{X_1 \cup X_3}(A', d) = \{\{w\}\}$, $RED_{X_1 \cup X_4}(A', d) = \{\{h\}, \{t, w\}, \{n\}\}$, $RED_{X_2 \cup X_3}(A', d) = \{\{t\}, \{h, w\}\}$, $RED_{X_2 \cup X_4}(A', d) = \{\{h\}, \{n\}\}$.

We obtain $\min(\dot{\bigcup}_{X \in \{X_1, X_2\}, Y \in \{X_3, X_4\}} RED_{X \cup Y}(A', d)) = \min(\min(RED_{X_1 \cup X_3}(A', d) \dot{\cup} RED_{X_1 \cup X_4}(A', d)) \dot{\cup} \min(RED_{X_2 \cup X_3}(A', d) \dot{\cup} RED_{X_2 \cup X_4}(A', d))) = \min(\{\{h, w\}, \{t, w\}, \{w, n\}\} \dot{\cup} \{\{t, h\}, \{t, n\}, \{h, w\}\}) = \{\{h, w\}, \{t, w, n\}\} = RED(A', d)$.

5 Data decomposition-based algorithms for attribute reduction

This section proposes data decomposition algorithms for attribute reduction that are defined based on the theorems from the previous section. The algorithms are evaluated and their features are discussed.

5.1 Attribute reduction algorithms

Attribute reduction of an information system is performed according to the schema presented in Fig. 1. Step I of the schema is performed by a simple algorithm *Create_loc_list*. The data size threshold is defined by the user and its maximum can be determined by the memory capacity limit. Steps II and III are performed by an algorithm *Compute_subred*. To compute the subreduct sets of middle subtables, the algorithm is called with default parameters. Step IV is done using an algorithm *Compute_red*.

Figure 2 presents the schema for performing attribute reduction of a decision table. The operations circled by the ellipse, i.e. step III, are presented in more detail in Fig. 3. An algorithm *Create_set_loc_list* is used to perform steps I, II, and III(i). An algorithm called *Compute_rel_red* uses *Compute_loc_red* to perform steps III(ii) and III(iii). It also conducts the remaining steps III(iv) and IV. The final relative reduct is computed based on the sum of partial sums of relative subreducts denoted by SS_{v_i, v_j} in Figs. 2 and 3.

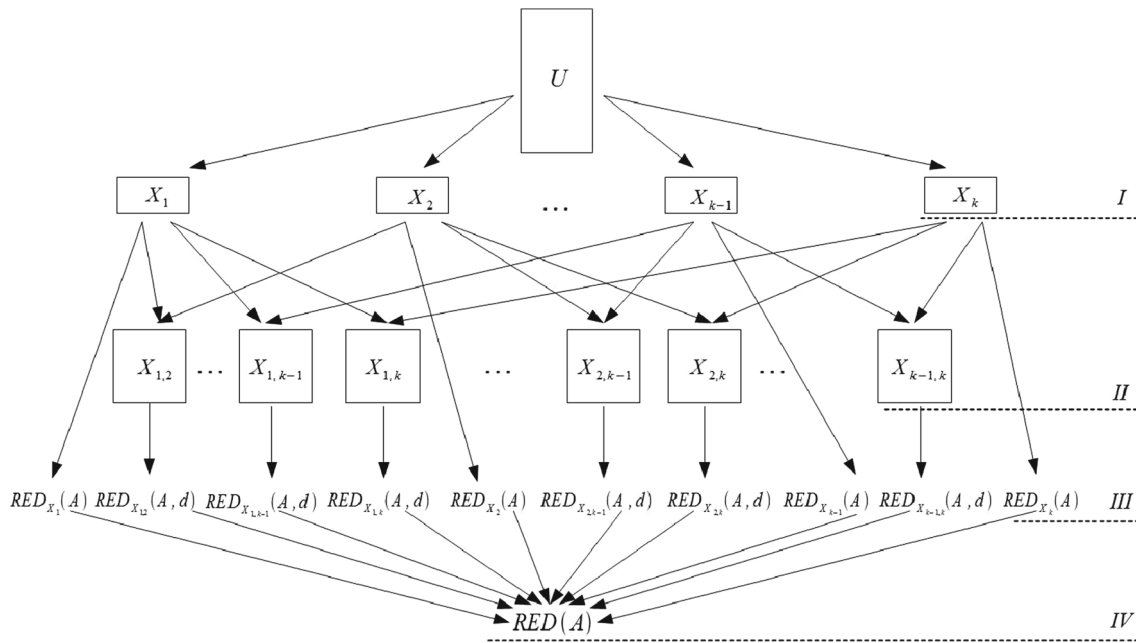


Fig. 1 Attribute reduction of $IS = (U, A)$: *I* Decomposition of the universe into middle subtables. *II* Construction of the final subtables based on the middle ones. *III* Computation of subreduct sets of the middle and final subtables. *IV* Merging subreduct sets into the reduct set

Fig. 2 Attribute reduction of $DT = (U, A \cup \{d\})$: *I* Decomposition of the universe into one-class subsets. *II* Construction of two-class subsets based on one-class ones. *III* Computation of relative subreduct sets of two-class subtables (partial sums of relative subreducts). *IV* Merging the partial sums of relative subreducts into the relative reduct set

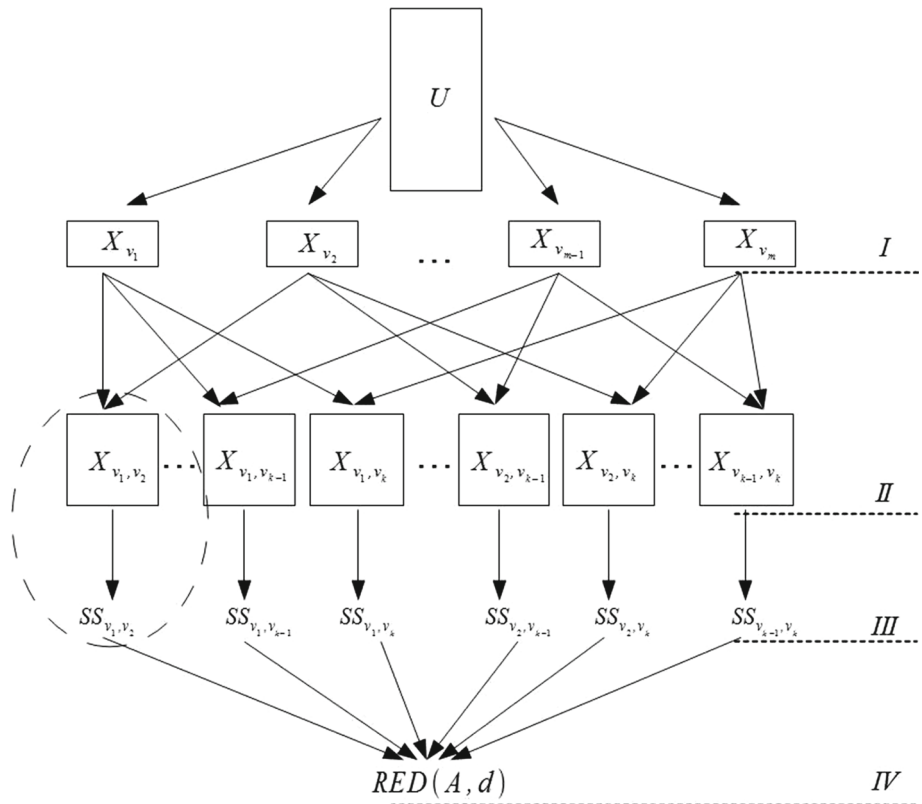
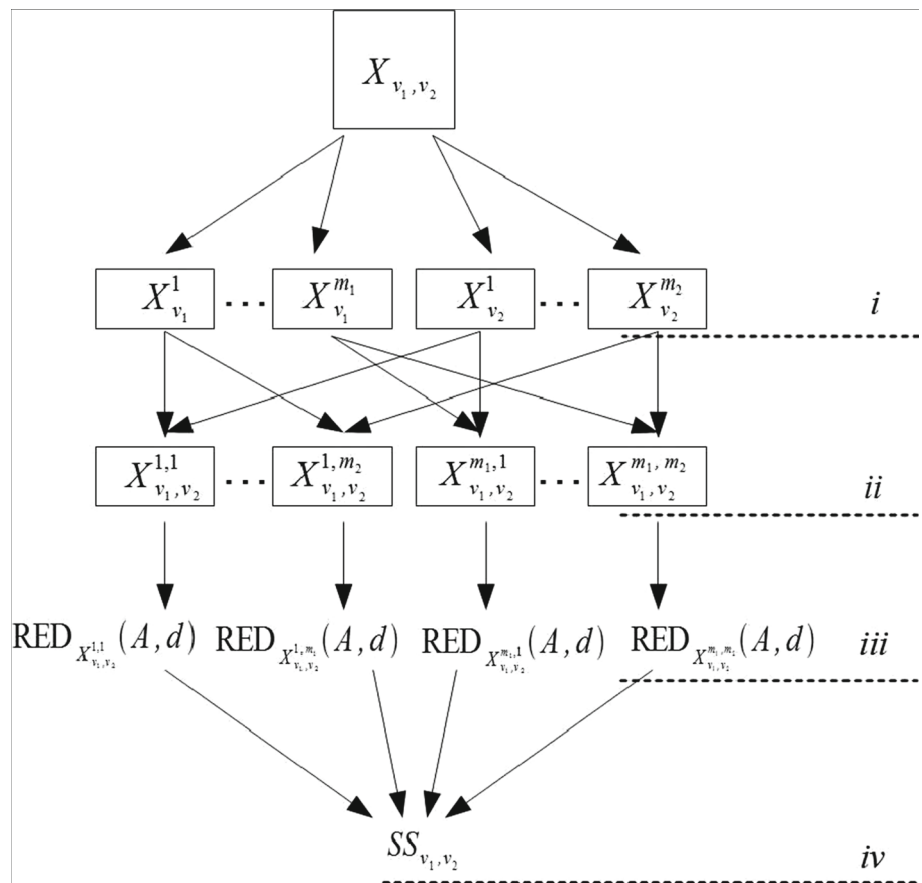


Fig. 3 Attribute reduction of two-class data subset: *i*
 Decomposition of each class of the two-class subtable into middle subtables. *ii*
 Construction of the final subtables based on the middle ones. *iii*
 Computation of the subreduct sets of the final subtables. *iv*
 Merging the relative subreduct sets into the partial sum of relative subreducts (SS_{v_1, v_2})



Algorithm 1: Create_loc_list

Data: X – a set of objects; m – the data size threshold
Result: L – a list of localizations of data subsets;

```

begin
  L := ∅;
  i := 1;
  while i ≤ |X| do
    L(i) := locate(X, i, m);
    /*returns the localization of the subset of X that includes
    m objects of X starting from the ith object */
    i := i + m;
  end
end
    
```

5.2 Evaluation of algorithms

The correctness and completeness of the proposed approach can be shown by Theorems 2 and 3.

Definition 14 (Correctness and completeness) A data decomposition-based algorithm for attribute reduction is correct and complete if for every data table using its every permissible decomposition, the algorithm returns the set of reducts of the whole data table.

Proposition 4 The Compute_red and Compute_rel_red algorithms are correct and complete.

Algorithm 2: Compute_subred

Data: L_1, L_2 – lists of localizations of data subsets; the default value for L_2 is L_1 ;
Result: SR – a set of subreducts;

```

begin
  LR := ∅;
  b := (L1 = L2);
  k1 = |L1|;
  if b = true then k2 := k1 else k2 := |L2| for i1 = 1 to k1 do
    X := load_data(L1(i1));
    /*loads the data identified by the localization L(i) */
    i := 1;
    if b = true then
      SR := SR ∪ {RED_X(A)};
      i := i + i1;
    end
    for i2 = i to k2 do
      Y := load_data(L2(i2));
      SR := SR ∪ {RED_{XUY}(A, d)};
    end
  end
end
    
```

The remaining of this section evaluates time and space complexity of the algorithms proposed in the previous subsection.

To create lists of localizations (the algorithms Create_loc_list and Create_set_loc_list), we need to once scan

Algorithm 3: *Compute_red*

Data: L – a list of localizations of data subsets;
Result: R – a set of reducts;
begin
 $SR := Compute_subred(L);$
 $R := \min(\dot{\bigcup}_{S \in SR} S);$
end

Algorithm 4: *Create_set_loc_list*

Data: X – a set of objects; d – the decision attribute; m – the data size threshold
Result: \mathcal{L} – a set of lists of localizations of data subsets;
begin
 $\mathcal{L} := \emptyset;$
 $i := 1;$
foreach $v \in V_d$ **do**
 $X' := class(X, v);$
*/*returns the subset of X that includes objects from class v */*
 $\mathcal{L}(i) := Create_loc_list(X', m);$
 $i := i + 1;$
end
end

Algorithm 5: *Compute_rel_red*

Data: \mathcal{L} – a set of lists of localizations of data subsets;
Result: R – a set of relative reducts;
begin
 $SR := \emptyset;$
 $k := |\mathcal{L}|;$
for $i_1 = 1$ **to** $k - 1$ **do**
 $L_1 := \mathcal{L}(i_1);$
for $i_2 = i_1 + 1$ **to** k **do**
 $L_2 := \mathcal{L}(i_2);$
 $SR := SR \cup Compute_subred(L_1, L_2);$
end
end
 $R := \min(\dot{\bigcup}_{S \in SR} S);$
end

the table, hence the time and space complexity of the algorithms is $O(n)$, where n is the number of objects stored in the table. Computing complexities, we do not include the data size defined by the number of attributes since this does not change for subtables.

The time complexity of the *Compute_red* and *Compute_rel_red* algorithms depends on the method to be used to compute subreducts. It will be shown that the time complexity of the *Compute_red* algorithm is equal to or less than that of the discernibility matrix based attribute reduction method when applied to the whole information system. To end this, one has to show that computations of reducts performed by *Compute_red* are equivalent to those of prime implicants of the discernibility function.

Based on the correspondence between the discernibility function of an information system and the set of reducts of this system one can define transformation operations.

Definition 15 (*Transformation operations*) Let $a, b \in A$ and $IS = (U, A)$ be an information system. Transformation operations are defined by

1. $a^* \wedge b^* \equiv \{\{a, b\}\},$
2. $a^* \vee b^* \equiv \{\{a\}, \{b\}\}.$

The following equivalencies can be derived from the above definition.

Proposition 5 Let $a, b, c \in A, B_i \subseteq A$ and $IS = (U, A)$ be an information system, where $1 \leq i \leq k$ and $k \geq 1$.

The following hold

- 5.1 $a^* \vee (b^* \wedge c^*) \equiv \{\{a\}, \{b, c\}\},$
- 5.2 $a^* \wedge (b^* \vee c^*) \equiv \{\{a, b\}, \{a, c\}\} = \{\{a\}\} \dot{\cup} \{\{b\}, \{c\}\},$
- 5.3 $\bigvee_{1 \leq i \leq k} \bigwedge_{a \in B_i} a^* \equiv \{\{a : a \in B_i : 1 \leq i \leq k\}\},$
- 5.4 $\bigwedge_{1 \leq i \leq k} \bigvee_{a \in B_i} a^* \equiv \dot{\bigcup}_{1 \leq i \leq k} \{\{a\} : a \in B_i\}.$

The reduct set computed for two different objects corresponds to the cell of discernibility matrix defined by the objects.

Proposition 6 Let $IS = (U, A)$ be an information system. The following holds

$$\forall_{\substack{x, y \in U, \\ x \neq y}} \left(\text{RED}_{\{x, y\}}(A) \equiv \bigvee_{a \in c_{x, y}} a^* \right) \tag{10}$$

The set constructed using the $\dot{\cup}$ operation on the reduct sets computed for all pairs of different objects corresponds to the discernibility function.

Proposition 7 Let $IS = (U, A)$ be an information system. The following holds

$$\dot{\bigcup}_{\substack{x, y \in U, \\ x \neq y}} \text{RED}_{\{x, y\}}(A) \equiv \bigwedge_{\substack{x, y \in U, \\ x \neq y}} \bigvee_{a \in c_{x, y}} a^* \tag{11}$$

Computing the left side of the equivalence (i.e. applying the $\dot{\cup}$ operation), we obtain an expression directly equivalent to disjunction form of the right side. This holds by the following proposition.

Proposition 8 Let $B_i \subseteq A$, where $1 \leq i \leq k$ and $k > 1$. The set $\dot{\bigcup}_{1 \leq i \leq k} \{\{a\} : a \in B_i\}$ in explicit form is directly equivalent to the expression $\bigwedge_{1 \leq i \leq k} \bigvee_{a \in B_i} a^*$ in disjunctive form.

Let $PI(p)$ be the set of all prime implicants of a Boolean expression p . The following three propositions show the equivalences of subsequent operations.

Proposition 9 Let $IS = (U, A)$ be an information system. The following holds

$$\min \left(\bigcup_{\substack{x,y \in U, \\ x \neq y}} \text{RED}_{\{x,y\}}(A) \right) \equiv \text{PI} \left(\bigwedge_{\substack{x,y \in U, \\ x \neq y}} \bigvee_{a \in C_{x,y}} a^* \right) \quad (12)$$

Proposition 10 Let $IS = (U, A)$ be an information system and $\{X_1, X_2, \dots, X_k\}$ be a covering of U , where $k > 1$. The following holds

$$\begin{aligned} & \bigcup_{1 \leq i < j \leq k} \min \left(\bigcup_{\substack{x,y \in U, \\ x \neq y}} \text{RED}_{\{x,y\}}(A) \right) \\ & \equiv \bigwedge_{1 \leq i < j \leq k} \text{PI} \left(\bigwedge_{\substack{x,y \in U, \\ x \neq y}} \bigvee_{a \in C_{x,y}} a^* \right) \end{aligned} \quad (13)$$

Proposition 11 Let $IS = (U, A)$ be an information system, $\{X_1, X_2, \dots, X_k\}$ be a covering of U ($k > 1$) and $IS_{ij} = (X_i \cup X_j, A)$ be an information subsystem of IS such that $X_i, X_j \subset U$. The following holds

$$\begin{aligned} & \min \left(\bigcup_{1 \leq i, j \leq k} \min \left(\bigcup_{\substack{x,y \in X_i \cup X_j, \\ x \neq y}} \text{RED}_{\{x,y\}}(A) \right) \right) \\ & \equiv \text{PI} \left(\bigwedge_{1 \leq i, j \leq k} \text{PI}(f_{IS_{ij}}(a_1^*, \dots, a_m^*)) \right) \end{aligned} \quad (14)$$

It is left to show that the left side of the above equivalence is the reduct set (1), and the right one is the set of prime implicants (2). Case (1) holds by the following corollary.

Corollary 1 (of Theorem 1) Let $IS = (U, A)$ be an information system and $\{X_1, X_2, \dots, X_k\}$ be a covering of U , where $k > 1$. The following holds

$$\begin{aligned} & \text{RED}(A) \\ & = \min \left(\bigcup_{1 \leq i < j \leq k} \min \left(\bigcup_{\substack{x,y \in X_i \cup X_j, \\ x \neq y}} \text{RED}_{\{x,y\}}(A) \right) \right) \end{aligned} \quad (15)$$

Case (2) holds by the following propositions.

Proposition 12 Let $IS = (U, A)$ be an information system, $\{X_1, X_2, \dots, X_k\}$ be a covering of U ($k > 1$) and $IS_{ij} = (X_i \cup X_j, A)$ be an information subsystem of IS such that $X_i, X_j \subset U$. The following holds

$$f_{IS}(a_1^*, \dots, a_m^*) \Leftrightarrow \bigwedge_{1 \leq i, j \leq k} f_{IS_{ij}}(a_1^*, \dots, a_m^*) \quad (16)$$

Proposition 13 Let $IS = (U, A)$ be an information system, $\{X_1, X_2, \dots, X_k\}$ be a covering of U ($k > 1$) and $IS_{ij} = (X_i \cup X_j, A)$ be an information subsystem of IS such that $X_i, X_j \subset U$. The following holds

$$\begin{aligned} & \text{PI}(f_{IS}(a_1^*, \dots, a_m^*)) \\ & \Leftrightarrow \text{PI} \left(\bigwedge_{1 \leq i, j \leq k} \text{PI}(f_{IS_{ij}}(a_1^*, \dots, a_m^*)) \right) \end{aligned} \quad (17)$$

One can analogously show that the time complexity of the *Compute_rel_red* algorithm is equal to or less than that of the relative discernibility matrix-based attribute reduction method applied to the whole decision table.

The space complexity of both the algorithms (i.e. *Compute_red* and *Compute_rel_red*) depends on the attribute reduction method used to compute subreducts and the size of data subset. One can estimate that the space complexity is between $O(m)$ and $O(m^2)$, where m is the size of data subset. It is assumed here that the data are processed in a batch mode, i.e. the whole data are placed in the memory. The space of the maximal size is needed when the discernibility matrix-based attribute reduction method is directly used and we have to store values of $\frac{m(m-1)}{2}$ matrix cells. One should note that for respectively small subtables the space complexity is determined by the number of possible reducts.

6 Experiments

This section describes experimental research that concerns attribute reduction of information systems using the approach proposed in this paper.

The approach was tested on 15 datasets taken from the UCI Repository (UCIRepository 2014). The discernibility matrix method was employed to compute reducts from decomposed data (datasets were decomposed into 4, 10, and 25 middle subtables) as well as from non-decomposed data for performance comparison. The approach was implemented in C++ and tested using a laptop (Intel Core i5, 2.3 GHz, 4 GB RAM, Windows 7).

Tables 1, 2, and 3 include the basic characteristic of each dataset (name, the number of attributes and objects), information on the number of reducts (for decomposed data, the average and the maximal (in brackets) number of subreducts are given), time taken to compute reducts (in seconds), and the ratio that expresses an increase or decrease of the time needed for computing reducts from decomposed data. A progress with respect to the non-decomposed data is written in bold. A progress with respect to the previous decomposition is written in italics.

One can observe that a shortening of the time needed for computing reducts occurs more often for datasets with fewer attributes. The shortening also clearly depends on the

Table 1 Attribute reduction for datasets with fewer than 10 attributes

Electricity board—5:45781				
No. mid. subtbl.	1	4	10	25
No. reducts	2	2.6(4)	3.30(4)	3.68(40)
Time taken	336.730	321.818	321.818	324.446
Ratio	1	0.955	0.955	0.963
Car evolution—7:1728				
No. mid. subtbl.	1	4	10	25
No. reducts	1	1(1)	1.29(3)	1.55(5)
Time taken	0.561	0.531	0.546	0.530
Ratio	1	0.95	0.97	0.94
Kingrook vs king (krk)—7:28056				
No. mid. subtbl.	1	4	10	25
No. reducts	1	1.5(2)	1.78(3)	2.11(6)
Time taken	148.003	141.228	144.754	138.731
Ratio	1	0.95	0.98	0.94
Pima Indians diabetes—9:768				
No. mid. subtbl.	1	4	10	25
No. reducts	27	25.4(32)	23.96(34)	20.38(31)
Time taken	0.187	0.203	0.234	0.533
Ratio	1	1.09	1.25	2.85
Nursery—9:12960				
No. mid. subtbl.	1	4	10	25
No. reducts	1	1(1)	2(1.22)	4(1.53)
Time taken	33.976	32.104	32.261	33.087
Ratio	1	0.94	0.95	0.97

Table 2 Attribute reduction for datasets with fewer than 20 attributes

Shuttle—10:43500				
No. mid. subtbl.	1	4	10	25
No. reducts	1	1(1)	1(1)	1.11(4)
Time taken	422.215	421.514	417.783	420.015
Ratio	1	0.998	0.990	0.995
Breast cancer Wisconsin—11:699				
No. mid. subtbl.	1	4	10	25
No. reducts	2	3.4(10)	2.35(10)	1.58(13)
Time taken	0.140	0.156	0.156	0.172
Ratio	1	1.11	1.11	1.23
Wine—14:178				
No. mid. subtbl.	1	4	10	25
No. reducts	128	68.3(80)	47.45(73)	19.0(51)
Time taken	0.125	0.218	0.327	0.374
Ratio	1	1.74	2.62	2.99
Australian credit approval—15:690				
No. mid. subtbl.	1	4	10	25
No. reducts	13	71.5(129)	88.8(154)	74.1(141)
Time taken	0.219	0.609	2.917	9.328
Ratio	1	2.78	13.32	42.6
Adult—15:32561				
No. mid. subtbl.	1	4	10	25
No. reducts	2	4.6(13)	11.8(28)	20.32(56)
Time taken	392.777	401.981	385.292	407.436
Ratio	1	1.02	0.98	1.04

number of reducts. If that number is very small (one or two reducts), then the computation time is more likely to be shorter. For datasets with bigger number of reducts other regularity is observed. Namely, a bigger number of middle subtables translates, in general, into a longer time needed for computing reducts. For some datasets, a small increase of the number of middle subtables can considerably lengthen the computation time. For example, for the Turkiye student evaluation dataset we can observe small time increases for 4 and 10 subtables and an over fourfold increase for 20 subtables. Furthermore, it is needed more than half an hour to compute the reducts for 25 subtables. The reason of this phenomenon can be searched in a big increase of the number of subreducts (see the maximal number of subreducts for the Australian credit approval, mushroom, and Turkiye student evaluation datasets).

7 Discussion

This section discusses features of the proposed approach.

An important feature is the reduction of space complexity. Namely, a table that does not fit in the memory can be

split into smaller data subtables. The size of subtables can be arbitrary small (i.e. at least two objects). However, the number of final subtables increases squaredly with respect to that of middle subtables (i.e. the increase is $\frac{k(k-1)}{2}$ for k middle subtables).

In spite of computations on many subtables, the time complexity of the approach does not have to increase compared with the direct attribute reduction. As shown, the approach in conjunction with the discernibility matrix-based attribute reduction method is equivalent to the direct attribute reduction of the whole table using this method. The difference is that in the former case subtasks (computations of subreducts) of the attribute reduction task are specified. This specification does not theoretically cause an increase in the number of computations. Thanks to this, the change of the size of subtables the table is to be split into does not influence the time complexity.

In practice, the number of computations can increase if the reducibility of the discernibility matrices of subtables is suitably smaller than that for the whole table. In such a case, we have more operations when computing and joining subreducts. The below example illustrates the problem of reducibility.

Table 3 Attribute reduction for datasets with more than 20 attributes

Meta-data—22:528				
No. mid. subt.	1	4	10	25
No. reducts	33	21.2(33)	22.27(41)	18.29(39)
Time taken	0.147	0.183	0.187	0.281
Ratio	1	1.24	1.27	1.91
Mushroom—23:8124				
No. mid. subt.	1	4	10	25
No. reducts	1	39.4(323)	154.9(1226)	216.5(1124)
Time taken	32.558	43.368	172.568	929.894
Ratio	1	1.33	5.30	28.56
Trains—33:10				
No. mid. subt.	1	4	10	25
No. reducts	3588	152.4(381)	13.1(24)	–
Time taken	26.471	22.451	26.541	–
Ratio	1	0.85	1.003	–
Turkiye student evaluation—33:5820				
No. mid. subt.	1	4	10	20*
No. reducts	1	98.2(724)	36.25(287)	61.1(2205)
Time taken	20.795	21.855	22.261	84.365
Ratio	1	1.05	1.07	4.06
Sonar, mines vs. rocks—61:208				
No. mid. subt.	1	4	10	25
No. reducts	1714	567.1(721)	174.38(311)	68.48(114)
Time taken	259.475	403.355	357.556	321.059
Ratio	1	1.55	1.38	1.24

Example 5 Let $X = \{1, 2, 6\}$. The discernibility matrix of (X, A) consists of cells $\{t, h, w, f\}$, $\{t, w, n\}$, $\{t, h, n, f\}$ and is not reducible. The part of the discernibility matrix of (U, A) defined by all pairs of objects from X reduces to \emptyset due to the cell $\{t\}$ that is included in other part of the discernibility matrix of (U, A) .

Another essential feature of the approach is that computations on subtables are performed independently from one another. Thanks to this, subreducts can be computed using any existing tool for attribute reduction.

The mentioned-above features causes the approach to be suitable for hardware implementation of attribute reduction (using programmable devices FPGAs). Hardware implementation can significantly speed up attribute reduction. Namely, thanks to parallelism, basic operations such as computation of discernibility matrix cells can be done in one cycle. This step was implemented and tested for the purpose of the computation of attribute core (Grzes et al. 2013). The memory capacity limitation of FPGA is not a barrier for large data processing since it can be split into suitably small portions. However, the division of a data table into many subtables can in practice extend the time needed for joining a huge number

of subreducts. Namely, for suitably large datasets the number of reducts of a smaller subtable is usually higher (see Tables 1, 2, 3).

8 Related works

This section compares the proposed approach with other rough set-based approaches that employ horizontal data decomposition for attribute reduction.

In Deng and Huang (2006) it was observed that the discernibility matrix of a decision table used for computing reducts can be represented by its submatrices. The universe is partitioned into subsets and each submatrix corresponds to one subset or to the sum of two subsets. The conjunction of the discernibility functions derived from the submatrices equals to that of the original matrix. The method enables to divide the decision table into arbitrary small subtables but it requires the use of a concrete attribute reduction method, i.e. one that is based on the discernibility matrix.

A decomposition and merging reduction method was proposed in Hu et al. (2006). For each of two subtables, formed by partitioning the universe, the reduct set is computed using any attribute reduction method. To obtain the final reduct set, the two reduct sets are merged by applying a method that is based on the notion of collision elementary set. This solution decreased the computation time compared with a standard attribute reduction method, but it is developed for two subtables only.

A core attributes based method proposed in Ye and Wu (2010) generates minimal reducts. The universe is divided into equivalence classes defined by the indiscernibility relation generated by the core attributes. Each subtable includes objects of one equivalence class and is limited to the attributes not belonging to the core. All possible unions of reducts (each reduct comes from a different subtable) are computed. The reducts of the minimal cardinality unions are expanded by the core attributes forming thereby the minimal reducts. Thanks to using the core for construction of subtables, the method reduces the number of both objects and attributes. However, the whole decision table is needed to compute the core and the solution can only be applied if the core is not empty.

The solution proposed in this paper is most similar to that from Deng and Huang (2006). Namely, the latter is almost a special case of the former that uses the discernibility matrix to compute reducts of a decision system. The difference is in the way the universe is partitioned. Each universe subset in the approach proposed in this paper includes objects from one class only. Thanks to this, the subsets do not need to be scan separately to compute relative reducts (only the sum of subsets are processed).

9 Conclusions

The problem of attribute reduction still needs new solutions to meet the challenges posed by large data.

This paper has proposed and investigated new definitions of attribute reduction using horizontal data decomposition. Algorithms for computing reducts of an information system as well as decision table have been developed and evaluated. The main benefits of the proposed approach are summarized as follows.

1. Data table can be split into arbitrarily small subtables (at least two objects).
2. Reduct sets of subtables are computed independently from one another using any standard attribute reduction method.
3. Compared with direct attribute reduction methods, the approach decreases the space complexity (it is determined by the size of subtables obtained during the decomposition) and achieves the same or less theoretical time complexity (it depends on the method used to compute reducts of subtables).
4. The time needed for computing reducts of information systems can be shorten when the data table includes fewer attribute (up to 10) or the number of reducts is small (one or two).

Future research includes more extensive experiments (the computation of reducts of decision tables, other methods for computing all reducts) and hardware implementation of the proposed method using FPGA devices.

Acknowledgments The project was funded by the National Science Center awarded on the basis of the Decision Number DEC-2012/07/B/ST6/01504.

Open Access This article is distributed under the terms of the Creative Commons Attribution License which permits any use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

Appendix

Appendix shows proofs of the propositions and theorems from Sects. 4 and 5.

Proof of Proposition 1 1.1–3 Straightforward from Definition 9.

- 1.4 The following holds
 $(\text{IND}(B) \Leftrightarrow \text{IND}(A)) \Leftrightarrow (\forall_{x,y \in U} \exists_{B' \in B} (x,y) \in \text{IND}(B') \Leftrightarrow (x,y) \in \text{IND}(A))$ (1).

By the assumption we have $\forall_{x,y \in X} (x,y) \in \text{IND}(B) \Leftrightarrow (x,y) \in \text{IND}(A)$. Let $B' \in A$ be such that $\forall_{\substack{x,y \in U \\ x \notin X \vee y \notin X}} (x,y) \in \text{IND}(B') \Leftrightarrow (x,y) \in \text{IND}(A)$. We obtain $\forall_{x,y \in U} \exists_{B'' \subseteq B \cup B'} (x,y) \in \text{IND}(B'') \Leftrightarrow (x,y) \in \text{IND}(A)$. By (1) we obtain $\text{IND}(B \cup B') \Leftrightarrow \text{IND}(A)$. Hence, $\exists_{C=B \cup B'} \text{IND}(C) \Leftrightarrow \text{IND}(A)$.

- 1.5 This can be shown analogously to 1.4. □

Proof of Proposition 2 2.1 We have $X \in \mathcal{S} \Rightarrow X = X \cup X \in \{Y \cup Y' : Y, Y' \in \mathcal{S}\} = \mathcal{S} \dot{\cup} \mathcal{S}$.

- 2.2 The following hold:
 $\min(\mathcal{S} \dot{\cup} \mathcal{S}') = \{X \cup X' : X \in \mathcal{S}, X' \in \mathcal{S}'\}$,
 $\forall_{Y \in \mathcal{S}, Y' \in \mathcal{S}'} Y \cup Y' \notin X \cup X'$ (1),
 $\forall_{X,Y} X \cup X, Y \cup Y \subseteq X \cup Y$ (2).
 We have $\min(\mathcal{S} \dot{\cup} \mathcal{S}) \stackrel{\text{by(1)}}{=} \{X \cup X' : X, X' \in \mathcal{S}\}$,
 $\forall_{Y,Y' \in \mathcal{S}} Y \cup Y' \notin X \cup X' \stackrel{\text{by(2)}}{=} \{X \cup X : X \in \mathcal{S}, \forall_{Y \in \mathcal{S}} Y \not\subseteq X\} = \min(\mathcal{S})$.

- 2.3 The following holds: $\mathcal{S} \subseteq \mathcal{S}', \mathcal{Z} \subseteq \mathcal{Z}' \Rightarrow \mathcal{S} \dot{\cup} \mathcal{Z} \subseteq \mathcal{S}' \dot{\cup} \mathcal{Z}'$ (1).

By (1) we have $\min(\mathcal{S}) \dot{\cup} \min(\mathcal{S}') \subseteq \mathcal{S}' \dot{\cup} \mathcal{S}' \Leftrightarrow \mathcal{S}' \dot{\cup} \mathcal{S}' = \min(\mathcal{S}) \dot{\cup} \min(\mathcal{S}') \cup \mathcal{Z}$, where $\mathcal{Z} = \{X \cup X' : X \in \mathcal{S}, X' \in \mathcal{S}', X \notin \min(\mathcal{S}) \vee X' \notin \min(\mathcal{S}')\}$. We obtain $\min(\mathcal{S}' \dot{\cup} \mathcal{S}') = \min(\min(\mathcal{S}) \dot{\cup} \min(\mathcal{S}') \cup \mathcal{Z})$. We need to show that $\min(\min(\mathcal{S}) \dot{\cup} \min(\mathcal{S}') \cup \mathcal{Z}) = \min(\min(\mathcal{S}) \dot{\cup} \min(\mathcal{S}'))$. Assume that the equality does not hold. We obtain $\exists_{X \cup X' \in \min(\min(\mathcal{S}) \dot{\cup} \min(\mathcal{S}') \cup \mathcal{Z})} X \cup X' \in \mathcal{Z}$ (2).

- (a) The case $X \notin \min(\mathcal{S})$. We have $X \in \mathcal{S} \wedge X \notin \min(\mathcal{S}) \Rightarrow \exists_{Y \in \min(\mathcal{S})} Y \subset X$ and $Y \cup X' \subset X \cup X'$. Hence, we have a contradiction with (2).
- (b) The case $X' \notin \min(\mathcal{S}')$. This can be shown analogously to the previous case.
- (c) The case $X \notin \min(\mathcal{S}) \wedge X' \notin \min(\mathcal{S}')$. This is the conjunction of the previous cases.

- 2.4 Let $\mathbf{S} = \{\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_k\}$ and $\mathbf{S}' = \{\mathcal{S}'_1, \mathcal{S}'_2, \dots, \mathcal{S}'_{k'}\}$, where $k, k' \geq 1$. We obtain $\dot{\bigcup}_{\mathcal{S} \in \mathbf{S} \cup \mathbf{S}'} \mathcal{S} = \mathcal{S}_1 \dot{\cup} \mathcal{S}_2 \dot{\cup} \dots \dot{\cup} \mathcal{S}_k \dot{\cup} \mathcal{S}'_1 \dot{\cup} \mathcal{S}'_2 \dot{\cup} \dots \dot{\cup} \mathcal{S}'_{k'}$
 $\dot{\bigcup}_{\mathcal{S} \in \mathbf{S}} \mathcal{S} \dot{\cup} \dot{\bigcup}_{\mathcal{S} \in \mathbf{S}'} \mathcal{S}$. □

Proof of Theorem 1 By Definition 4 we have $\text{RED}(A) = \text{PI}(f_{\text{IS}}(a_1^*, \dots, a_m^*))$. □

Lemma 1 Let $\text{IS} = (U, A)$ be an information system. The following holds $\text{PI}(f_{\text{IS}}(a_1^*, \dots, a_m^*)) \Leftrightarrow \min(\mathcal{S}_{f_{\text{IS}}})$, where $\mathcal{S}_{f_{\text{IS}}}$ is the family of sets constructed according to Definition 15 and Proposition 5 and corresponding to f_{IS} .

Proof It is enough to show that f_{IS} after the application of the absorption law is equivalent to $\mathcal{S}_{f_{\text{IS}}}$ after the application of the min function.

Since f_{IS} is given in conjunctive normal form, then the following absorption law is used: $a^* \wedge (a^* \vee b^*) \Leftrightarrow a^*$, where $a, b \in A$. Let $al(p)$ return an expression that is equivalent to p after the application of the absorption law. We have $a^* \wedge (a^* \vee b^*) \equiv \{\{a\}, \{a, b\}\}$ by Proposition 5 and $p(a^* \wedge (a^* \vee b^*)) = a^* \equiv \{\{a\}\} = \min(\{\{a\}, \{a, b\}\})$. It can be shown analogously for any Boolean expression in conjunctive form. \square

We obtain

1. $RED_{X_i \cup X_j}(A) \Leftrightarrow PI(f_{i,j}(a_1^*, \dots, a_m^*))$ for any $X_i, X_j \subseteq U$ by Definition 4,
2. $\dot{\bigcup}_{1 \leq i, j \leq k} RED_{X_i \cup X_j}(A) \Leftrightarrow \bigwedge_{1 \leq i, j \leq k} PI(f_{i,j}(a_1^*, \dots, a_m^*))$ by Proposition 5,
3. $\min(\dot{\bigcup}_{1 \leq i, j \leq k} RED_{X_i \cup X_j}(A)) \Leftrightarrow PI(\bigwedge_{1 \leq i, j \leq k} PI(f_{i,j}(a_1^*, \dots, a_m^*)))$ by the lemma,
4. $PI(\bigwedge_{1 \leq i, j \leq k} PI(f_{i,j}(a_1^*, \dots, a_m^*))) = PI(f_{IS}(a_1^*, \dots, a_m^*))$ by Proposition 13.

Proof of Theorem 2 We need to show that $\min(\dot{\bigcup}_{1 \leq i < j \leq k} RED_{X_i \cup X_j}(A)) = \min(\dot{\bigcup}_{1 \leq i \leq k} RED_{X_i}(A) \dot{\cup} \dot{\bigcup}_{1 \leq i < j \leq k} RED_{X_i \cup X_j}(A, d))$.

Lemma 2 Let $IS = (U, A)$ be an information system. The following holds $\forall X, Y \subseteq U RED_{X \cup Y}(A) = \min(RED_X(A) \dot{\cup} RED_Y(A) \dot{\cup} RED_{X \cup Y}(A, d))$.

Proof We obtain $RED_{X \cup Y}(A) = \min(\dot{\bigcup}_{x, y \in X \cup Y} RED_{\{x\} \cup \{y\}}(A))$ by Theorem 1, where the covering of $X \cup Y$ is $\{\{x\} : x \in X \cup Y\}$. We have $\dot{\bigcup}_{x, y \in X \cup Y} RED_{\{x\} \cup \{y\}}(A) = \dot{\bigcup}_{x \in X \cup Y} RED_{\{x\} \cup \{y\}}(A) = \dot{\bigcup}_{x \in X} RED_{\{x\} \cup \{y\}}(A) \dot{\cup} \dot{\bigcup}_{x \in Y} RED_{\{x\} \cup \{y\}}(A) \dot{\cup} \dot{\bigcup}_{x \in X} RED_{\{x\} \cup \{y\}}(A) \dot{\cup} \dot{\bigcup}_{x \in Y} RED_{\{x\} \cup \{y\}}(A)$ (A) by Proposition 5 and the property $(S \cup S') \times (S \cup S') = (S \times S) \cup (S' \times S') \cup (S' \times S) \cup (S \times S')$.

Sublemma 4 Let $IS = (U, A)$ be an information system. The following holds $\forall X, Y \subseteq U (\dot{\bigcup}_{x \in X, y \in Y} RED_{\{x\} \cup \{y\}}(A) = \dot{\bigcup}_{x, y \in X \cup Y} RED_{\{x\} \cup \{y\}}(A, d))$.

- Proof* 1. The case $x = y$. We obtain $RED_{\{x\} \cup \{y\}}(A) = RED_{\{x\} \cup \{y\}}(A, d)$ by Definition 7.
2. The case $x \neq y$.

1. The case $d(x) = d(y)$. We obtain $\neg(x \in X \wedge y \in Y)$, hence $RED_{\{x\} \cup \{y\}}(A)$ is not computed. This is equivalent to $RED_{\{x\} \cup \{y\}}(A) = \emptyset$. We also obtain $RED_{\{x\} \cup \{y\}}(A, d) = \emptyset$ by Definition 7.
2. The case $d(x) \neq d(y)$. We obtain $RED_{\{x\} \cup \{y\}}(A) = RED_{\{x\} \cup \{y\}}(A, d)$ by Definition 7.

(end of sublemma proof)

For simplicity's sake we denote $B_1 = \dot{\bigcup}_{y \in X} RED_{\{x\} \cup \{y\}}$ (A), $B_2 = \dot{\bigcup}_{y \in Y} RED_{\{x\} \cup \{y\}}$ (A), $B_3 = \dot{\bigcup}_{x \in X} RED_{\{x\} \cup \{y\}}$ (A), $B_4 = \dot{\bigcup}_{y \in Y} RED_{\{x\} \cup \{y\}}$ (A). We obtain $\min(B_1 \dot{\cup} B_2 \dot{\cup} B_3 \dot{\cup} B_4) = \min(\min(B_1) \dot{\cup} \min(B_2) \dot{\cup} \min(B_3 \dot{\cup} B_4))$ by Properties 2.3 and 2.4. We have $B_3 = B_4$ by the symmetry of $RED_{\{x\} \cup \{y\}}(A)$ with respect to x and y . Hence, $\min(B_3 \dot{\cup} B_4) = \min(B_3)$ by 2.3 and 2.4. We have by the sublemma that $B_3 = \dot{\bigcup}_{x, y \in X \cup Y} RED_{\{x\} \cup \{y\}}(A, d) = B_5$. Therefore, we have $\min(\min(B_1) \dot{\cup} \min(B_2) \dot{\cup} \min(B_3 \dot{\cup} B_4)) = \min(\min(B_1) \dot{\cup} \min(B_2) \dot{\cup} \min(B_5)) = \min(RED_X(A) \dot{\cup} RED_Y(A) \dot{\cup} RED_{X \cup Y}(A, d))$ by Theorem 1. (end of lemma proof).

By the lemma we obtain $\min(\dot{\bigcup}_{1 \leq i < j \leq k} RED_{X_i \cup X_j}(A)) = \min(\dot{\bigcup}_{1 \leq i < j \leq k} \min(RED_{X_i}(A) \dot{\cup} RED_{X_j}(A) \dot{\cup} RED_{X_i \cup X_j}(A, d)))$. Let $\mathcal{B}_i = RED_{X_i}(A)$, $\mathcal{B}_j = RED_{X_j}(A)$, $\mathcal{B}_{ij} = RED_{X_i \cup X_j}(A, d)$. We have $\min(\dot{\bigcup}_{1 \leq i < j \leq k} \min(\mathcal{B}_i \dot{\cup} \mathcal{B}_j \dot{\cup} \mathcal{B}_{ij})) = \min(\dot{\bigcup}_{1 \leq i < j \leq k} (\mathcal{B}_i \dot{\cup} \mathcal{B}_j \dot{\cup} \mathcal{B}_{ij})) = \min(\dot{\bigcup}_{1 \leq i < j \leq k} (\mathcal{B}_i \dot{\cup} \mathcal{B}_j) \dot{\cup} \dot{\bigcup}_{1 \leq i < j \leq k} \mathcal{B}_{ij}) = \min(\min(\dot{\bigcup}_{1 \leq i < j \leq k} (\mathcal{B}_i \dot{\cup} \mathcal{B}_j)) \dot{\cup} \min(\dot{\bigcup}_{1 \leq i < j \leq k} \mathcal{B}_{ij}))$.

Lemma 3 Let $IS = (U, A)$ be an information system, $\{X_1, X_2, \dots, X_k\}$ ($k > 1$) be a covering of U . The following holds $\min(\dot{\bigcup}_{1 \leq i < j \leq k} RED_{X_i}(A) \dot{\cup} RED_{X_j}(A)) = \min(\dot{\bigcup}_{1 \leq i \leq k} RED_{X_i}(A))$.

Proof We obtain $\min(\dot{\bigcup}_{1 \leq i < j \leq k} RED_{X_i}(A) \dot{\cup} RED_{X_j}(A)) = \min(RED_{X_1}(A) \dot{\cup} RED_{X_2}(A) \dot{\cup} RED_{X_2}(A) \dot{\cup} \dots \dot{\cup} RED_{X_{k-1}}(A) \dot{\cup} RED_{X_{k-1}}(A) \dot{\cup} RED_{X_k}(A) \dot{\cup} RED_{X_k}(A)) = \min(\min(RED_{X_1}(A)) \dot{\cup} \min(RED_{X_2}(A) \dot{\cup} RED_{X_2}(A)) \dot{\cup} \dots \dot{\cup} \min(RED_{X_{k-1}}(A) \dot{\cup} RED_{X_{k-1}}(A)) \dot{\cup} \min(RED_{X_k}(A) \dot{\cup} RED_{X_k}(A))) = \min(\min(RED_{X_1}(A)) \dot{\cup} \min(RED_{X_2}(A)) \dot{\cup} \dots \dot{\cup} \min(RED_{X_{k-1}}(A)) \dot{\cup} \min(RED_{X_k}(A))) = \min(RED_{X_1}(A) \dot{\cup} RED_{X_2}(A) \dot{\cup} \dots \dot{\cup} RED_{X_{k-1}}(A) \dot{\cup} RED_{X_k}(A)) = \min(\dot{\bigcup}_{1 \leq i \leq k} RED_{X_i}(A))$.

(end of lemma proof).

By the lemma we have $\min(\min(\dot{\bigcup}_{1 \leq i < j \leq k} (\mathcal{B}_i \dot{\cup} \mathcal{B}_j)) \dot{\cup} \min(\dot{\bigcup}_{1 \leq i < j \leq k} \mathcal{B}_{ij})) = \min(\min(\dot{\bigcup}_{1 \leq i \leq k} \mathcal{B}_i) \dot{\cup} \min(\dot{\bigcup}_{1 \leq i < j \leq k} \mathcal{B}_{ij})) = \min(\dot{\bigcup}_{1 \leq i \leq k} \mathcal{B}_i \dot{\cup} \dot{\bigcup}_{1 \leq i < j \leq k} \mathcal{B}_{ij}) = \min(\dot{\bigcup}_{1 \leq i \leq k} RED_{X_i}(A) \dot{\cup} \dot{\bigcup}_{1 \leq i < j \leq k} RED_{X_i \cup X_j}(A, d))$.

Proposition 3 and Theorem 3 can be proved analogously to Proposition 1 and Theorem 1, respectively.

Proof of Proposition 4 Straightforward by Theorems 2 and 3.

Proof of Proposition 5.1 By Definition 15.
 5.2 By the distribution of conjunction over disjunction law and Definition 15.

5.3 By 5.1.

5.4 By 5.2.

Proof of Proposition 6 We have $\bigvee_{a \in c_{x,y}} a^* \equiv \{\{a\} : a \in c_{x,y}\} = \{\{a\} : a \in A, a(x) \neq a(y)\}$. Therefore, we need to show that $\forall_{x,y \in U, x \neq y} (B \in \text{RED}_{\{x,y\}}(A) \Leftrightarrow \exists_{a \in A} B = \{a\} \wedge a(x) \neq a(y))$.

1. The case “ \Rightarrow ”. Assume that $B \in \text{RED}_{\{x,y\}}(A) \wedge (\forall_{a \in A} B \neq \{a\}(1) \vee a(x) = a(y)(2))$.

We have $x \neq y \Leftrightarrow \exists_{a \in A} a(x) \neq a(y)$ (3) and $B \in \text{RED}_{\{x,y\}}(A) \Leftrightarrow \text{IND}_{\{x,y\}}(B) = \text{IND}_{\{x,y\}}(A)$ (4).

By (3) and (4) we have $(x, y) \notin \text{IND}_{\{x,y\}}(A) \wedge (x, y) \notin \text{IND}_{\{x,y\}}(B)$, hence $\exists_{a \in B} a(x) \neq a(y)$. This leads to a contradiction with (2).

We obtain $\exists_{a \in B} a(x) \neq a(y) \Rightarrow \forall_{C \subseteq A, a \in C} \text{IND}_{\{x,y\}}(C) = \text{IND}_{\{x,y\}}(\{a\})$. Hence, we obtain $B = \{a\}$ and have a contradiction with (1).

2. The case “ \Leftarrow ”. Assume that $\exists_{a \in A} B = \{a\} \wedge a(x) \neq a(y) \wedge B \notin \text{RED}_{\{x,y\}}(A)$.

We have $B \notin \text{RED}_{\{x,y\}}(A)$, hence

(a) The case $\exists_{C \subseteq B, C \neq \emptyset} C \in \text{RED}_{\{x,y\}}(A)$. We have $|B| = 1 \Rightarrow \forall_{D \subseteq B} D = \emptyset$, hence we have a contradiction with Definition 7.

(b) The case $\exists_{C \supset B} C \in \text{RED}_{\{x,y\}}(A)$. We obtain $a \in B \wedge a(x) \neq a(y) \Rightarrow \forall_{D \supseteq B} \text{IND}_{\{x,y\}}(D) = \text{IND}_{\{x,y\}}(B)$. Hence, we obtain $C \notin \text{RED}_{\{x,y\}}(A)$ and have a contradiction with the assumption.

Proof of Proposition 7 Straightforward by Propositions 5 and 6.

Proof of Proposition 8 Let $B_i = \{a_1^i, a_2^i, \dots, a_{l_i}^i\}$, where $l_i \geq 1$ and $1 \leq i \leq k$.

We obtain $L = \bigcup_{1 \leq i \leq k} \{\{a\} : a \in B_i\} = \bigcup_{1 \leq i \leq k} \{\{a_1^i\}, \{a_2^i\}, \dots, \{a_{l_i}^i\}\} = \{\{a_{j_1}^1, a_{j_2}^2, \dots, a_{j_k}^k\} : 1 \leq j_1 \leq l_1, 1 \leq j_2 \leq l_2, \dots, 1 \leq j_k \leq l_k\}$ by Definition 11.

We also obtain $R = \bigwedge_{1 \leq i \leq k} \bigvee_{a \in B_i} a^* \Leftrightarrow \bigwedge_{1 \leq i \leq k} (a_1^{i*} \vee a_2^{i*} \vee \dots \vee a_{l_i}^{i*}) \Leftrightarrow \bigvee_{1 \leq j_1 \leq l_1} \bigvee_{1 \leq j_2 \leq l_2} \dots \bigvee_{1 \leq j_k \leq l_k} (a_{j_1}^{1*} \wedge a_{j_2}^{2*} \wedge \dots \wedge a_{j_k}^{k*})$ by the distributive law.

Finally, we obtain $L \equiv R$ by Proposition 5.

Proof of Proposition 9 It can be shown analogously to Lemma 1.

Proof of Proposition 10 Straightforward by Propositions 5 and 9.

Proof of Proposition 11 Straightforward by Propositions 9 and 10.

Proof of Proposition 12 Let $\{X_1, X_2, \dots, X_k\}$ be a covering of U . The following holds $U \times U = \bigcup_{1 \leq i, j \leq k} X_i \times X_j$ (1).

We obtain $f_{IS}(a_1^*, \dots, a_m^*) = \bigwedge_{x,y \in U} \bigvee_{a \in c_{x,y}} a^* \Leftrightarrow \bigwedge_{(x,y) \in U \times U} \bigvee_{a \in c_{x,y}} a^* \stackrel{\text{by (1)}}{\Leftrightarrow} \bigwedge_{1 \leq i, j \leq k} \bigwedge_{(x,y) \in X_i \times X_j} \bigvee_{a \in c_{x,y}} a^* \Leftrightarrow \bigwedge_{1 \leq i, j \leq k} f_{IS_{i,j}}(a_1^*, \dots, a_m^*)$.

Proof of Proposition 13 Computation of prime implicants of a Boolean function is equivalent to application of the absorption laws to the function.

Lemma 4 Let $\bigwedge_{i=1}^k p_i \Leftrightarrow \bigwedge_{i=1}^{k_1} p_i \wedge \bigwedge_{i=k_1+1}^{k_2} p_i \wedge \dots \wedge \bigwedge_{i=k_l+1}^k p_i$, where p_i is a disjunction of atomic formulas ($1 \leq i \leq k$), $k > 1$ and $l < k$. The following holds $al(\bigwedge_{i=1}^k p_i) = al(al(\bigwedge_{i=1}^{k_1} p_i) \wedge al(\bigwedge_{i=k_1+1}^{k_2} p_i) \wedge \dots \wedge al(\bigwedge_{i=k_l+1}^k p_i))$.

Proof An operation al_{\wedge} is defined based on the absorption law as follows

$$p \text{ al}_{\wedge} q = \begin{cases} p, & \text{if } q \Rightarrow p; \\ q, & \text{if } p \Rightarrow q; \\ p \wedge q, & \text{otherwise.} \end{cases}$$

The following holds $\forall_{p,q,r} p \text{ al}_{\wedge} q \text{ al}_{\wedge} r \Leftrightarrow (p \text{ al}_{\wedge} q) \text{ al}_{\wedge} r \Leftrightarrow p \text{ al}_{\wedge} (q \text{ al}_{\wedge} r)$ (1). We obtain $L = al(\bigwedge_{i=1}^k p_i) = p_1 \text{ al}_{\wedge} p_2 \text{ al}_{\wedge} \dots \text{ al}_{\wedge} p_k$ by (1). We also obtain $R = al(al(\bigwedge_{i=1}^{k_1} p_i) \wedge al(\bigwedge_{i=k_1+1}^{k_2} p_i) \wedge \dots \wedge al(\bigwedge_{i=k_l+1}^k p_i)) = al((p_1 \text{ al}_{\wedge} p_2 \text{ al}_{\wedge} \dots \text{ al}_{\wedge} p_{k_1}) \wedge (p_{k_1+1} \text{ al}_{\wedge} p_{k_1+2} \text{ al}_{\wedge} \dots \text{ al}_{\wedge} p_{k_2}) \wedge \dots \wedge (p_{k_l+1} \text{ al}_{\wedge} p_{k_l+2} \text{ al}_{\wedge} \dots \text{ al}_{\wedge} p_k)) = (p_1 \text{ al}_{\wedge} p_2 \text{ al}_{\wedge} \dots \text{ al}_{\wedge} p_{k_1}) \text{ al}_{\wedge} (p_{k_1+1} \text{ al}_{\wedge} p_{k_1+2} \text{ al}_{\wedge} \dots \text{ al}_{\wedge} p_{k_2}) \text{ al}_{\wedge} \dots \text{ al}_{\wedge} (p_{k_l+1} \text{ al}_{\wedge} p_{k_l+2} \text{ al}_{\wedge} \dots \text{ al}_{\wedge} p_k)$ by (1). Hence, we have $L = R$.

(end of lemma proof)

By the lemma we obtain $al(f_{IS}(a_1^*, \dots, a_m^*)) = al(\bigwedge_{1 \leq i, j \leq k} al(f_{IS_{i,j}}(a_1^*, \dots, a_m^*)))$.

References

Bazan J, Nguyen HS, Nguyen SH, Synak P, Wróblewski J (2000) Rough set methods and applications. In: Rough set algorithms in classification problem. Physica-Verlag GmbH, Heidelberg, pp 49–88

Bazan JG, Skowron A, Synak P (1994) Dynamic reducts as a tool for extracting laws from decisions tables. In: Ras ZW, Zemankova M (eds) ISMIS. Lecture Notes in Computer Science, pp 346–355

Chen D, Zhao S, Zhang L, Yang Y, Zhang X (2012) Sample pair selection for attribute reduction with rough set. IEEE Trans Knowl Data Eng 24(11):2080–2093

Degang C, Changzhong W, Qinghua H (2007) A new approach to attribute reduction of consistent and inconsistent covering decision systems with covering rough sets. Inf Sci 177(17):3500–3518

Deng D, Huang H (2006) A new discernibility matrix and function. In: RSKT, Lecture Notes in Computer Science, vol 4062, pp 114–121

Grzes T, Kopczynski M, Stepaniuk J (2013) FPGA in rough set based core and reduct computation. In: Lingras P, Wolski M, Cornelis C, Mitra S, Wasilewski P (eds) RSKT. Lecture Notes in Computer Science, vol 8171. Springer, Berlin, pp 263–270

Grzymala-Busse J (1991) An algorithm for computing a single covering. Kluwer Academic Publishers, Dordrecht, p 66

- Hu F, Fan X, Yang SX, Tao C (2006) A novel reduction algorithm based decomposition and merging strategy, vol 344. Lecture notes in control and information sciences. Springer, Berlin, pp 790–796
- Hu X, Cercone N (1995) Learning in relational databases: a rough set approach. *Comput Intell* 11(2):323–338
- Kryszkiewicz M (1998) Rough set approach to incomplete information systems. *Inf Sci* 112(1–4):39–49
- Kryszkiewicz M (2001) Comparative study of alternative type of knowledge reduction in inconsistent systems. *Int J Intell Syst* 16:105–120
- Leung Y, Li D (2003) Maximal consistent block technique for rule acquisition in incomplete information systems. *Inf Sci* 153(1):85–106
- Li H, Zhu J (2005) Finding all the absolute reductions based on discernibility matrix. In: Proceedings of 2005 international conference on machine learning and cybernetics, IEEE, vol 9, pp 5682–5685
- Miao D, Zhao Y, Yao Y, Li HX, Xu F (2009) Relative reducts in consistent and inconsistent decision tables of the Pawlak rough set model. *Inf Sci* 179(24):4140–4150
- Nguyen SH, Skowron A, Synak P (1998) Discovery of data patterns with applications to decomposition and classification problems. In: Polkowski L, Skowron A (eds) *Rough sets in knowledge discovery 2: applications, case studies and software systems*. Studies in fuzziness and soft computing, vol 19, Physica-Verlag, Heidelberg, chap 4, pp 55–97
- Pawlak Z (1991) *Rough sets. Theoretical aspects of reasoning about data*. Kluwer Academic, Dordrecht
- Skowron A, Rauszer C (1992) The discernibility matrices and functions in information systems. In: *Intelligent decision support*. Springer, Amsterdam, pp 331–362
- Ślęzak D (1999) Decomposition and synthesis of decision tables with respect to generalized decision functions. In: Pal S, Skowron A (eds) *Rough fuzzy hybridization: a new trend in decision making*. Springer, Berlin, pp 110–135
- Ślęzak D (2002) Approximate entropy reducts. *Fundam Inf* 53(3–4):365–390
- Starzyk J, Nelson DE, Sturtz K (1999) Reduct generation in information systems. In: *Bulletin of international rough set society*, vol 3, issue 1–2. pp 19–22
- Suraj Z (1996) Discovery of concurrent data models from experimental tables: a rough set approach. *Fund Inf* 28(3–4):353–376
- Susmaga R (1998) Effective tests for inclusion minimality in reduct generation. *Found Comput Decis Sci* 4(23):219–240
- Swiniarski R (2001) Rough sets methods in feature reduction and classification. *Int J Appl Math Comput Sci* 11(3):565–582
- Thi VD, Giang NL (2013) A method for extracting knowledge from decision tables in terms of functional dependencies. *Cybern Inf Technol* 13(1):73–82
- Tsang E, Degang C, Yeung D (2008) Approximations and reducts with covering generalized rough sets. *Comput Math Appl* 56(1):279–289
- UCI Repository (2014). <http://archive.ics.uci.edu/ml/>. Accessed 18 Nov 2014
- Wang Y, Ma L (2009) FF-based feature selection for improved classification of medical data. *WSEAS Trans Comput* 8(2):396–405
- Xu Z, Huang L, Qian W, Yang B (2009) Quick attribute reduction algorithm based on improved frequent pattern tree. In: *IEEE international conference on intelligent computing and intelligent systems*, IEEE, vol 1, pp 406–410
- Yang M (2006) An incremental updating algorithm of the computation of a core based on the improved discernibility matrix. *Chin J Comput* 29(3):407–413
- Ye D, Chen Z (2002) A new discernibility matrix and the computation of a core. *Acta Electron Sin* 30(7):1086–1088
- Ye M, Wu C (2010) Decision table decomposition using core attributes partition for attribute reduction. In: *5th international conference on computer science and education (ICCSE)*, IEEE, vol 23, pp 23–26
- Zhang WX, Mi JS, Wu WZ (2003) Approaches to knowledge reductions in inconsistent systems. *Int J Intell Syst* 18(9):989–1000
- Zhang X, Mei C, Chen D, Li J (2013) Multi-confidence rule acquisition oriented attribute reduction of covering decision systems via combinatorial optimization. *Knowl Based Syst* 50:187–197