Y.Y. Yao and Y. Zhao,
Discernibility matrix simplification for constructing attribute reducts,
Information Sciences, Vol. 179, No. 5, 867-882, 2009.

Discernibility Matrix Simplification for Constructing Attribute Reducts

Yiyu Yao and Yan Zhao

Department of Computer Science, University of Regina,
Regina, Saskatchewan, Canada S4S 0A2

Abstract

This paper proposes a reduct construction method based on discernibility matrix simplification. The method works in a similar way to the classical Gaussian elimination method for solving a system of linear equations. Elementary matrix simplification operations are introduced. Each operation transforms a matrix into a simpler form. By applying these operations a finite number of times, one can transform a discernibility matrix into one of its minimum (i.e., the simplest) forms. Elements of a minimum discernibility matrix are either the empty set or singleton subsets, in which the union derives a reduct. With respect to an ordering of attributes, which is either computed based on a certain measure of attributes or directly given by a user, two heuristic reduct construction algorithms are presented. One algorithm attempts to exclude unimportant attributes from a reduct, and the other attempts to include important attributes in a reduct.

Key words: rough sets, attribute reduction, discernibility matrix, matrix simplification, algorithms

1 Introduction

Data analysis, dependency analysis, and learning are some of the most important applications of rough set theory. In those applications, it is typically

Email address: {yyao, yanzhao}@cs.uregina.ca (Yiyu Yao and Yan Zhao).

assumed that we have a finite set of objects described by a finite set of attributes. The values of objects on attributes can be conveniently represented by an information table, with rows representing objects and columns representing attributes. The notion of a reduct plays an essential role in analyzing an information table [19,21]. An attribute reduct is a minimal subset of attributes that provides the same descriptive ability as the entire set of attributes. In other words, attributes in a reduct is jointly sufficient and individually necessary. We obtain the same results and conclusions by using a smaller set of attributes. As a consequence, we normally obtain more general and simpler rules.

Many methods have been proposed and examined for finding the set of all reducts or a single reduct [1,15–17,20,23,26,27,29,33]. Unfortunately, it has been proven that finding the set of all reducts, or finding an optimal reduct (i.e., a reduct with the minimum number of attributes), is NP [31]. Thus, many heuristic methods for finding one reduct have also been investigated [2,3,6,8–10,14,28,30,32,34–36]. Software has been developed for reduct construction. For example, RSES is a collection of algorithms and data structures for rough set computations [25]. Several results from the study of reducts are particularly relevant to the present paper.

A beautiful theoretical result was developed by Skowron and Rauszer [26], based on the notion of a discernibility matrix. Both the rows and columns of the matrix correspond to the objects. An element of the matrix is the set of all attributes that distinguish the corresponding object pairs, namely, the set consists of all attributes on which the corresponding two objects have distinct values. One can construct a Boolean discernibility function from a discernibility relation, with attributes as Boolean variables. Skowron and Rauszer [26] showed that the set of attribute reducts are in fact the set of prime implicants of the reduced disjunctive form of the discernibility function. This provides a logic foundation for the study of reducts. Many researchers studied reduct construction by using the discernibility information in the discernibility matrix [4,12,18,20,22,30,34]. However, a difficulty exists when applying the theoretical results. It is not very clear how to design an efficient algorithm for constructing *one* reduct based on manipulating the discernibility function.

Pawlak [20] introduced the notion of a minimum discernibility matrix. Given a reduct, a minimum discernibility matrix of the reduct is a matrix whose elements are either the empty set or singleton subsets. An element is the empty set if the original element is the empty set. Otherwise, it is obtained by selecting one attribute in the reduct from the original element. The union of all elements in the minimum matrix is the reduct that produces the minimum matrix. This property of a minimum matrix has a valuable implication. If we could find a systematic method for simplifying a discernibility matrix to a minimum discernibility matrix, we would have a reduct construction method.

Unfortunately, an algorithm does not exist yet for producing a minimum matrix.

Wang and his colleagues [30,33,34] proposed an elegant algorithm for constructing a reduct by using the discernibility matrix and an ordering of attributes. The algorithm is based on a set representation of the discernibility matrix. That is, the discernibility matrix is transformed into a set of subsets of attributes corresponding to the elements of the matrix. A few operations on the set representation are introduced. The absorption operation deletes all elements that are supersets of other elements. Suppose attributes in each subset are ordered according to the given ordering. One can immediately introduce the grouping operation that produces a partition of all elements based on the very first attribute of each element. The algorithm starts from the empty set as a partial reduct, i.e., a subset of a reduct. Based on the results of absorption and grouping, an attribute is selected to be added to a constructed partial reduct, and an attribute subset is deleted from all elements in the set representation. With the modified set representation, the algorithm repeats the above steps until a reduct is constructed. A detailed description of the algorithm can be found in [34]. Unfortunately, this elegant algorithm has not received its due attention from researchers in the rough set community.

By combining the results from these studies, we introduce a method for matrix simplification in order to find a reduct. Elementary matrix operations are presented. They are similar to the operations used by Wang and his colleagues [34]. Each operation produces a simpler discernibility matrix. Based on these operations, we present a method for transforming a discernibility matrix. The working principle of our method is in the same spirit of the classical Gaussian elimination method for solving a system of linear equations. By applying the elementary operations a finite number of times, we obtain one minimum discernibility matrix. The union of all elements in the minimum matrix produces a reduct.

In order to demonstrate the usefulness and flexibility of the proposed method, we consider an ordering of attributes representing their importance. Two heuristic algorithms are suggested. One algorithm involves deleting unimportant attributes and the other involves keeping important attributes.

The main objective of this paper is to present a theoretical result regarding systematic discernibility matrix simplification, although some heuristics are discussed for obtaining efficient algorithms. Based on the basic results of the paper, it is possible to study a wide class of matrix simplification algorithms.

2 Reducts in an Information Table

In many data analysis applications, information and knowledge are stored and represented in an *information table*. An information table provides a convenient way to describe a finite set of objects within a universe by a finite set of attributes [19].

Definition 1 (Information table) *An information table is the following tuple:*

$$S = (U, At, \{V_a \mid a \in At\}, \{I_a \mid a \in At\}),$$

where U is a finite non-empty set of objects, At is a finite non-empty set of attributes, V_a is a non-empty set of values of $a \in At$, and $I_a : U \longrightarrow V_a$ is an information function that maps an object of U to exactly one value in V_a.

An information table represents all available information and knowledge. That is, objects are only perceived, observed, or measured by using a finite number of attributes.

Given a subset of attributes $A \subseteq At$, an indiscernibility relation $IND(A) \subseteq U \times U$ is defined by:

$$IND(A) = \{(x, y) \in U \times U \mid \forall a \in A, I_a(x) = I_a(y)\}.$$

For any two objects $x, y \in U$, if $x\ IND(A)\ y$, then x and y are indiscernible based on the attribute set A.

According to the indiscernibility relation, Pawlak defined a reduct in an information table as a minimum set of attributes that keeps the indiscernibility relation $IND(At)$ unchanged [20].

Definition 2 (Reduct) *Given an information table S, an attribute set $R \subseteq At$ is called a reduct, if R satisfies the two conditions:*

(i). *$IND(R) = IND(At)$;*
(ii). *For any $a \in R$, $IND(R - \{a\}) \neq IND(At)$.*

The first condition indicates the joint sufficiency of the attribute set R. The object pairs that cannot be distinguished (indiscernible) by R still cannot be distinguished (indiscernible) by At, and vice versa. The second condition indicates that each attribute in R is individually necessary. There exist object pairs that cannot be distinguished (indiscernible) by $R - \{a\}$ but can be distinguished by At. It means that R is the minimum attribute set that can keep the indiscernibility relation $IND(At)$. Normally, there is more than one reduct in an information table. The set of all reducts of an information table S is denoted as $RED(S)$.

A special type of information tables can be denoted as $S = (U, At = C \cup \{D\}, \{V_a \mid a \in At\}, \{I_a \mid a \in At\})$, where C is a set of conditional attributes and D is a decision attribute indicating the decision class of each object in the universe. Such an information table is also called a *decision table*. The relative indiscernibility relation of two objects is then defined as:

$$\begin{aligned} IND(A|\{D\}) &= \{(x, y) \in U \times U \mid \forall a \in A \ [I_a(x) = I_a(y)] \vee I_D(x) = I_D(y)\} \\ &= IND(A) \cup IND(\{D\}). \end{aligned}$$

The relative indiscernibility relation consists of all object pairs that cannot be distinguished (indiscernible) based on the set A of conditional attributes or share the same value on the decision attribute D. Based on the relative indiscernibility relation, a relative reduct can be similarly defined [20].

3 The Discernibility Matrix and the Discernibility Function

This section reviews the relevant results of the discernibility matrix and the discernibility function [26]. They not only provide a convenient notation, but also lead to a logical foundation and a systematic way to compute the set of all reducts.

3.1 The discernibility matrix

Two objects are discernible if their values are different in at least one attribute. Skowron and Rauszer suggested a matrix representation for storing the sets of attributes that discern pairs of objects, called a *discernibility matrix* [26].

Definition 3 (Discernibility matrix) *Given an information table S, its discernibility matrix $M = (M(x, y))$ is a $|U| \times |U|$ matrix, in which the element $M(x, y)$ for an object pair (x, y) is defined by:*

$$M(x, y) = \{a \in At \mid I_a(x) \neq I_a(y)\}. \tag{1}$$

The physical meaning of the matrix element $M(x, y)$ is that objects x and y can be distinguished by any attribute in $M(x, y)$. The pair (x, y) can be discerned if $M(x, y) \neq \emptyset$. A discernibility matrix M is symmetric, i.e., $M(x, y) = M(y, x)$, and $M(x, x) = \emptyset$. Therefore, it is sufficient to consider only the lower triangle or the upper triangle of the matrix.

Table 1
An information table

U	a	b	c	d	e
1	high	middle	middle	fer1	pes1
2	middle	middle	middle	fer2	pes1
3	middle	high	weak	fer1	pes2
4	middle	high	middle	fer1	pes1
5	low	middle	strong	fer1	pes1
6	high	middle	middle	fer3	pes2
7	high	high	middle	fer1	pes2

By the definitions of indiscernibility relation and discernibility matrix, we have:

$$\begin{aligned} x\ IND(\{a\})\ y &\Longleftrightarrow a \notin M(x,y), \\ x\ IND(A)\ y &\Longleftrightarrow A \cap M(x,y) = \emptyset. \end{aligned}$$

Therefore, they can be defined by each other as follows:

$$\begin{aligned} IND(A) &= \{(x,y) \in U \times U \mid A \cap M(x,y) = \emptyset\}; \\ M(x,y) &= \{a \in At \mid \neg(x\ IND(\{a\})\ y)\}. \end{aligned}$$

An attribute set $A \subseteq At$ can discern an object pair (x,y) if $A \cap M(x,y) \neq \emptyset$.

Example 1 *We adopt the information table shown in Table 1 from* [34] *to illustrate the basic concepts, where a, b, c, d and e represent temperature, humidity, wind, fertilization and pesticide, respectively.*

The discernibility matrix of Table 1 is shown below. For the underlined object pair (2,1), the entry $\{a, d\}$ indicates that either attribute a or d discerns the two objects.

Table 2
A decision table

U	a	b	c	d	e	D
1	high	middle	middle	fer1	pes1	high
2	middle	middle	middle	fer2	pes1	high
3	middle	high	weak	fer1	pes2	high
4	middle	high	middle	fer1	pes1	low
5	low	middle	strong	fer1	pes1	low
6	high	middle	middle	fer3	pes2	low
7	high	high	middle	fer1	pes2	low

$$\begin{bmatrix} & & & & & & \\ \underline{\{a,d\}} & & & & & & \\ \{a,b,c,e\} & \{b,c,d,e\} & & & & & \\ \{a,b\} & \{b,d\} & \{c,e\} & & & & \\ \{a,c\} & \{a,c,d\} & \{a,b,c,e\} & \{a,b,c\} & & & \\ \{d,e\} & \{a,d,e\} & \{a,b,c,d\} & \{a,b,d,e\} & \{a,c,d,e\} & & \\ \{b,e\} & \{a,b,d,e\} & \{a,c\} & \{a,e\} & \{a,b,c,e\} & \{b,d\} & \end{bmatrix}$$

For a decision table, the relative discernibility can also be stored in a matrix form in which an element is defined as:

$$M(x,y) = \{a \in C \mid [I_a(x) \neq I_a(y)] \wedge [I_D(x) \neq I_D(y)]\}. \qquad (2)$$

That is, we only consider the discernibility of objects in different decision classes.

Example 2 *The decision table shown in Table 2, adopted from* [34]*, can be used to illustrate a relative discernibility matrix. The discernibility matrix is shown below.*

$$\begin{bmatrix} & & & & & & \\ \emptyset & & & & & & \\ \emptyset & \emptyset & & & & & \\ \{a,b\} & \{b,d\} & \{c,e\} & & & & \\ \{a,c\} & \{a,c,d\} & \{a,b,c,e\} & \emptyset & & & \\ \{d,e\} & \{a,d,e\} & \{a,b,c,d\} & \emptyset & \emptyset & & \\ \{b,e\} & \{a,b,d,e\} & \{a,c\} & \emptyset & \emptyset & \emptyset & \end{bmatrix} \tag{3}$$

Once a discernibility matrix is constructed, it can be processed in the same way, independent of whether it is a normal discernibility matrix or a relative discernibility matrix. In the rest of the paper, we consider them uniformly as a discernibility matrix.

3.2 The discernibility function

From a discernibility matrix, one can define the notion of a discernibility function [26].

Definition 4 (Discernibility function) *The discernibility function of a discernibility matrix is defined by:*

$$f(M) = \bigwedge\{\bigvee(M(x,y)) | \forall x, y \in U, M(x,y) \neq \emptyset\}. \tag{4}$$

The expression $\bigvee(M(x,y))$ is the disjunction of all attributes in $M(x,y)$, indicating that the object pair (x,y) can be distinguished by any attribute in $M(x,y)$. The expression $\bigwedge\{\bigvee(M(x,y))\}$ is the conjunction of all $\bigvee(M(x,y))$, indicating that the family of discernible object pairs can be distinguished by a set of attributes satisfying $\bigwedge\{\bigvee(M(x,y))\}$.

The discernibility function can be used to state an important result regarding the set of reducts of an information table, as shown by the following theorem from Skowron and Rauszer [26].

Theorem 1 *The reduct set problem is equivalent to the problem of transforming the discernibility function to a reduced disjunctive form. Each conjunctor of the reduced disjunctive form is called a prime implicant. Given the discernibility matrix M of an information table S, an attribute set $R = \{a_1, \ldots, a_p\}$*

is a reduct if and only if the conjunction of all attributes in R, denoted as $a_1 \wedge \cdots \wedge a_p$, is a prime implicant of $f(M)$.

In order to derive the reduced disjunctive form, the discernibility function $f(M)$ is transformed by using the absorption and distribution laws. Accordingly, finding the set of reducts can be modelled based on the manipulation of a Boolean function. The set $RED(S)$ of reducts of an information table is equivalent to the set of prime implicants of the discernibility function.

Based on the results of Theorem 1, Skowron and Rauszer [26] also suggested an alternative characterization of a reduct in terms of the discernibility matrix as shown by the next theorem.

Theorem 2 *Given the discernibility matrix M of an information table S, an attribute set R is a reduct if and only if*

(i). $\forall (x, y) \in U \times U \ [M(x, y) \neq \emptyset \Longrightarrow R \cap M(x, y) \neq \emptyset]$;
(ii). $\forall a \in R \ \exists (x, y) \in U \times U \ [M(x, y) \neq \emptyset \ \wedge \ ((R - \{a\}) \cap M(x, y) = \emptyset)]$.

Property (i) shows that R is jointly sufficient for distinguishing all discernible object pairs. In fact, the set of attributes formed by the union of all elements of the discernibility matrix satisfies property (i). Property (ii) shows that each attribute in R is individually necessary. The result of Theorem 2 provides a convenient way to test if a subset of attributes is a reduct. However, it does not directly offer a method to compute a reduct. Many authors have proposed and studied various algorithms to construct a reduct based on the discernibility matrix [18,20,22,30,34]. For simplicity, we denote the set of reducts of an information table in terms of the discernibility matrix by $RED(M)$.

4 Discernibility Matrix Simplification

In his book on rough sets, Pawlak [20] introduced the notion of a minimum discernibility matrix. He noted that the union of all elements of the minimum matrix is a reduct.

4.1 Minimum discernibility matrix

Given a reduct R, for all $M(x, y) \neq \emptyset$, we have $R \cap M(x, y) \neq \emptyset$. Thus, for each non-empty $M(x, y)$, we can replace it by one and only one attribute in R to preserve the original discernibility of x and y. Such a simplest form is called a minimum discernibility matrix [20]. For a given reduct R, we may obtain more than one minimum discernibility matrix.

Example 3 *Consider Table 1. It can be verified that attribute sets $\{a,b,e\}$ and $\{a,b,c,d\}$ are two reducts. We can obtain the following two minimum discernibility matrices of the reducts $\{a,b,e\}$*

$$\begin{bmatrix} & & & & & \\ \{a\} & & & & & \\ \{a\} & \{b\} & & & & \\ \{a\} & \{b\} & \{e\} & & & \\ \{a\} & \{a\} & \{a\} & \{a\} & & \\ \{e\} & \{a\} & \{a\} & \{a\} & \{a\} & \\ \{e\} & \{a\} & \{a\} & \{a\} & \{a\} & \{b\} \end{bmatrix}$$

and $\{a,b,c,d\}$,

$$\begin{bmatrix} & & & & & \\ \{d\} & & & & & \\ \{b\} & \{b\} & & & & \\ \{b\} & \{b\} & \{c\} & & & \\ \{c\} & \{c\} & \{b\} & \{b\} & & \\ \{d\} & \{d\} & \{b\} & \{b\} & \{c\} & \\ \{b\} & \{b\} & \{c\} & \{a\} & \{b\} & \{b\} \end{bmatrix}$$

respectively.

The intuitive notion of a minimum discernibility matrix can be defined as follows.

Definition 5 (Minimum discernibility matrix) *Given an information table S with the discernibility matrix M, a matrix $M_0 = (M_0(x,y))$ is called a minimum discernibility matrix of M if M_0 satisfies the following conditions:*

(1.) *Any element of M_0 is either the empty set or a singleton attribute set;*
(2.) $\forall(x,y) \in U \times U\ [M(x,y) \neq \emptyset \Longrightarrow (M_0(x,y) \neq \emptyset \wedge M_0(x,y) \subseteq M(x,y))]$;
(3.) $\forall(x,y) \in U \times U\ [M(x,y) = \emptyset \Longrightarrow M_0(x,y) = \emptyset]$;
(4.) $\bigcup(M_0(x,y))$ *is a reduct.*

According to the definition, an element $M_0(x,y)$ of M_0 must be a subset of the corresponding element $M(x,y)$ of M. Furthermore, by conditions (2) and

(3), M and M_0 share the same qualitative property, namely, $M(x,y) \neq \emptyset \Longleftrightarrow M_0(x,y) \neq \emptyset$. The matrix M_0 may be viewed as a simplification of M. Since each element of M_0 is either the empty set or a singleton subset, M_0 may be viewed as the simplest form of M. By the fact that the union of all elements in M_0 is a reduct, M_0 gives an alternative representation of a reduct.

The procedure for constructing a minimum matrix from a reduct is trivial and of no practical value. On the other hand, the reverse procedure is useful. If we can transform M into one of its minimum forms, we can easily obtain a reduct by taking the union of all elements of the minimum matrix. However, an algorithm does not exist yet for constructing such a minimum matrix.

Intuitively speaking, an element $M_0(x,y)$ of the minimum matrix M_0 is obtained by deleting some attributes from $M(x,y)$ of M. This deletion operation may be viewed as a matrix simplification operation. In order to ensure that the simplified matrix contains a reduct, we need to establish criteria for deletion. That is, we need to study the conditions under which an attribute in each $M(x,y) \in M$ can be deleted, as well as systematic procedures of matrix simplification. In the rest of this paper, we present such a method.

4.2 Equivalence, specification and similarity of matrices

In order to precisely define matrix simplification operations, we first introduce several relationships between two discernibility matrices.

Definition 6 (Simplified discernibility matrix) *A matrix M_s is called a simplified discernibility matrix of a discernibility matrix M if the following conditions hold:*

(1.) $\forall (x,y) \in U \times U \; [M(x,y) \neq \emptyset \Longrightarrow (M_s(x,y) \neq \emptyset \wedge M_s(x,y) \subseteq M(x,y))]$;
(2.) $\forall (x,y) \in U \times U \; [M(x,y) = \emptyset \Longrightarrow M_s(x,y) = \emptyset]$.

In this paper, our discussion is restricted to simplified discernibility matrices. When no confusion arises, we also call a simplified discernibility matrix a discernibility matrix.

We can define three types of relations between two discernibility matrices. Two discernibility matrices are equivalent if they have exactly the same set of reducts. One is a specification of the other if its set of reducts is a subset of the set of reducts of the other. They are similar if they share at least one reduct. Formally, we have the following definitions.

Definition 7 (Equivalence, specification and similarity of matrices) *A discernibility matrix M_1 is equivalent to another discernibility matrix M_2,* *denoted as* $M_1 \equiv M_2$*, if* $RED(M_1) = RED(M_2)$*.*

A discernibility matrix M_1 *is a specification of another discernibility matrix* M_2*, or* M_2 *is a generalization of* M_1*, denoted as* $M_1 \preceq M_2$*, if* $RED(M_1) \subseteq RED(M_2)$*.*

A discernibility matrix M_1 *is similar to another discernibility matrix* M_2*, denoted as* $M_1 \approx M_2$*, if* $RED(M_1) \cap RED(M_2) \neq \emptyset$*.*

We can easily verify that

$$\begin{aligned} &\text{(i).} \quad M_1 \equiv M_2 \Longleftrightarrow M_1 \preceq M_2 \wedge M_2 \preceq M_1; \\ &\text{(ii).} \quad M_1 \equiv M_2 \Longrightarrow M_1 \approx M_2. \end{aligned} \tag{5}$$

The relation $\equiv$ is an equivalence relation, $\preceq$ is reflexive and transitive, and $\approx$ is reflexive and symmetric. For the purpose of constructing a minimum discernibility matrix, we need to ensure that the matrix simplification process is guided by one of these relations.

4.3 Elementary matrix operations

Based on the concepts of matrix equivalence and specification, we introduce elementary matrix operations for matrix simplification. While the absorption operations produce an equivalent matrix, the deletion operations produce a specification of the original matrix.

Element absorption: For a matrix element $M(x', y') \neq \emptyset$, it absorbs another element $M(x, y)$ if the following condition holds:

$$\emptyset \neq M(x', y') \subset M(x, y).$$

That is, the value of $M(x, y)$ is replaced by the value of $M(x', y')$ in the matrix. We also say $M(x, y)$ is absorbed by $M(x', y')$.

The physical meaning of element absorption can be explained as follows. Suppose $M(x', y') \neq \emptyset$ and $M(x', y') \subset M(x, y)$. The set of attributes discerning both pairs (x', y') and (x, y) is given by $M(x, y) \cap M(x', y') = M(x', y')$. After absorption, $M(x, y)$ becomes $M(x', y')$. Attributes in $M(x', y')$ are sufficient to discern both object pairs (x', y') and (x, y). When an attribute from $M(x', y')$ is in a reduct, the same attribute can be used to discern (x, y). Thus, it is not necessary to consider attributes in $M(x, y) - M(x', y')$.

Based on element absorption, Wang *et al.* [34] propose a generalized version of the matrix absorption operation. Specifically, one applies the element absorption operation for all element pairs in the matrix to obtain a simpler matrix.

Matrix absorption: The matrix absorption operation is a sequence of all possible element absorption operations on pairs of elements whenever the following condition holds:

$$\emptyset \neq M(x', y') \subset M(x, y).$$

After matrix absorption, no element in the matrix is a proper subset of another element.

It should be pointed out that the resulting matrix from matrix absorption is not unique and depends on the ordering in which different pairs of elements are absorbed. Nevertheless, all the absorbed matrices are equivalent to the original discernibility matrix.

Lemma 1 *The matrix obtained by the application of an element absorption operation is equivalent to the original matrix. A matrix obtained by the matrix absorption operation is equivalent to the original matrix.*

Proof. The element absorption operation picks an element $M(x_1, y_1) \neq \emptyset$ of M and works on another element $M(x_2, y_2)$ with $M(x_1, y_1) \subset M(x_2, y_2)$. Let M_e denote the matrix after the element absorption operation regarding the absorber $M(x_1, y_1)$. By the definition of element absorption, we have:

$$M_e(x, y) = \begin{cases} M(x_1, y_1), & (x, y) = (x_2, y_2); \\ M(x, y), & \text{otherwise.} \end{cases} \tag{6}$$

To prove the first part of the lemma, we need to show that $RED(M) \equiv RED(M_e)$.

$RED(M) \subseteq RED(M_e)$: Suppose $R \in RED(M)$. By Theorem 2, we have:

(i). $\forall (x, y) \in U \times U \; [M(x, y) \neq \emptyset \Longrightarrow R \cap M(x, y) \neq \emptyset]$;
(ii). $\forall a \in R \; \exists (x, y) \in U \times U \; [M(x, y) \neq \emptyset \; \wedge \; ((R - \{a\}) \cap M(x, y) = \emptyset)]$.

We now show that R also satisfies (i) and (ii) for the matrix M_e. According to Equation (6), for (i) we only need to show that $M_e(x_2, y_2) \cap R \neq \emptyset$. This is obviously the case because $M_e(x_2, y_2) = M(x_1, y_1)$ and $M(x_1, y_1) \cap R \neq \emptyset$. By the fact that $M_e(x, y) \subseteq M(x, y)$ for all object pairs and R satisfies (ii) for M, R satisfies (ii) for M_s. According to Theorem 2, $R \in RED(M_e)$. Thus, $RED(M) \subseteq RED(M_e)$.

$RED(M_e) \subseteq RED(M)$: Suppose $R \in RED(M_e)$. By Theorem 2, R satisfies (i) and (ii) for the matrix M_e. By the fact that $M_e(x, y) \subseteq M(x, y)$ for all object pairs, it follows that R satisfies (i) for M. Since R satisfies (ii) for M_e, we know that for any attribute $a \in R$ there exists $(x, y) \in U \times U$, such that $M_e(x, y) \neq \emptyset$ and $(R - \{a\}) \cap M_e(x, y) = \emptyset$. If $(x, y) \neq (x_2, y_2)$, we can immediately conclude that $M(x, y) = M_e(x, y) \neq \emptyset$ and $(R - \{a\}) \cap M(x, y) = (R - \{a\}) \cap M_e(x, y) = \emptyset$. If $(x, y) = (x_2, y_2)$, it follows $M_e(x_2, y_2) = M(x_1, y_1) \neq \emptyset$ and $(R - \{a\}) \cap M(x_1, y_1) = (R - \{a\}) \cap M_e(x_2, y_2) = \emptyset$. Therefore, if R satisfies (ii) for M_e, it satisfies (ii) for M. By Theorem 2, $R \in RED(M)$. Thus, $RED_e(M) \subseteq RED(M)$.

Since the number of elements of a matrix is finite, the above proof can be easily extended to the matrix absorption operation to prove the second part of the lemma. □

Attribute deletion: For an attribute $a \in At$, the attribute deletion operation deletes a from all the elements if the following condition holds:

$$\forall (M(x, y) \neq \emptyset)\ [(M(x, y) - \{a\}) \neq \emptyset].$$

In other words, attribute a can be deleted if $\{a\}$ is not a singleton set in M.

The physical meaning of attribute deletion can be seen from the result of Theorem 2. Recall that the union of all elements in the discernibility matrix, i.e., $W = \bigcup(M(x, y))$, satisfies the joint sufficiency property:

(i). $\quad \forall (x, y) \in U \times U\ [M(x, y) \neq \emptyset \Longrightarrow W \cap M(x, y) \neq \emptyset]$.

However, it may not satisfy the individual necessity property:

(ii). $\quad \forall a \in W\ \exists (x, y) \in U \times U\ [M(x, y) \neq \emptyset\ \wedge\ ((W - \{a\}) \cap M(x, y) = \emptyset)]$.

If the attribute set W does not satisfy property (ii), there must exist an attribute $a \in W$ such that

$$\forall (M(x, y) \neq \emptyset)\ [(W - \{a\}) \cap M(x, y) \neq \emptyset].$$

It follows that $W - \{a\}$ still satisfies property (i) and hence it can be deleted from W.

In general, one can apply the attribute deletion operation several times to delete a set of attributes.

Attribute set deletion: For an attribute set $A \subseteq At$, the attribute set deletion operation deletes all the attributes in A from all the elements if the following condition holds:

$$\forall (M(x, y) \neq \emptyset)\ [(M(x, y) - A) \neq \emptyset].$$

In other words, the attribute set A can be deleted if none of the subsets of A is an element of M.

There are several reasons to study the set deletion operation. First, the set deletion operation can be interpreted as a sequence of the single attribute deletion operations. It provides a short-hand notation for the sequence. Second, set deletion may reduce the computational cost. Third, in many reduct construction algorithms, set deletion plays an important role. This will be explained in detail in Section 5.

The results of the two deletion operations are unique, and the simplified matrices are specifications of the original discernibility matrix.

Lemma 2 *The matrix obtained by the application of an attribute deletion operation or an attribute set deletion operation is a specification of the original matrix.*

Proof. Let M_s denote the matrix after the attribute d is deleted by the attribute deletion operation. By the definition of attribute deletion, we have:

$$M_s(x,y) = \begin{cases} M(x,y), & d \notin M(x,y); \\ M(x,y) - \{d\}, & \text{otherwise.} \end{cases} \tag{7}$$

We need to show that $RED(M_s) \subseteq RED(M)$. Suppose $R \in RED(M_s)$. By Theorem 2, R satisfies (i) and (ii) for the matrix M_s. It follows that $d \notin R$, otherwise it does not satisfy the individual necessity property (ii). By the fact that $M_s(x,y) \subseteq M(x,y)$ for all object pairs, it follows that R satisfies (i) for M. Since R satisfies (ii) for M_s, we know that for any attribute $a \in R$ there exists $(x,y) \in U \times U$, such that $M_s(x,y) \neq \emptyset$ and $(R - \{a\}) \cap M_s(x,y) = \emptyset$. If $d \notin M(x,y)$, we can immediately conclude that $M(x,y) \neq \emptyset$ and $(R - \{a\}) \cap M(x,y) = \emptyset$. If $d \in M(x,y)$, from the fact that $d \notin R$, it follows $M(x,y) = M_s(x,y) \cup \{d\} \neq \emptyset$ and $(R - \{a\}) \cap (M_s(x,y) \cup \{d\}) = \emptyset$. Therefore, if R satisfies (ii) for M_s, it satisfies (ii) for M. By Theorem 2, $R \in RED(M)$. Thus, $RED_s(M) \subseteq RED(M)$.

Since the attribute set deletion operation can be interpreted as a finite number of attribute deletions, the same conclusion holds for the attribute set deletion operation. □

The resulting matrix from a deletion operation is a simplified matrix of the original matrix, as defined by Definition 6. As shown by Lemma 3, such a simplified matrix contains at least one reduct of the original matrix.

Lemma 3 *The matrix obtained by the application of an attribute deletion operation or an attribute set deletion operation contains at least one reduct of* *the original matrix.*

Proof. Suppose M_s is the simplified matrix by deleting an attribute $d \in At$ from the original discernibility matrix M. The union of all elements in M_s is $W = \bigcup(M_s(x,y)) = \bigcup(M(x,y)) - \{d\}$. From the fact that $M_s(x,y) \subseteq M(x,y)$, it follows that W satisfies the joint sufficiency property (i) of Theorem 2 for both matrices M and M_s. That is, W is a superset of a reduct, namely, a super-reduct [33], for both M and M_s. Hence, they must share at least one reduct, which is a subset of W. Similarly, we can prove that the simplified matrix after applying the attribute set deletion operation also must contain at least one reduct of the original matrix. □

Both the absorption and the deletion operations produce a simplified matrix. The following theorem shows that these operations are indeed sufficient to produce a minimum matrix.

Theorem 3 *A discernibility matrix can be transformed to a minimum matrix by the application of a finite number of elementary matrix operations.*

Proof. We first need to prove that a finite number of elementary matrix operations will produce a matrix which contains only the empty set or singleton subsets. Suppose there exists an element which contains at least two attributes, and let us denote it by $M(x_0, y_0)$. We consider two cases: (1) there exists a non-empty element of M which is a proper subset of $M(x_0, y_0)$, and (2) there does not exist a non-empty element of M which is a proper subset of $M(x_0, y_0)$. For case (1), the element $M(x_0, y_0)$ is absorbed by the proper subset, and hence its cardinality is reduced by at least one. For case (2), for any attribute $a \in M(x_0, y_0)$, $\{a\}$ is not a singleton attribute set in M. One can conclude that a satisfies the following condition for the attribute deletion operation:

$$\forall (M(x,y) \neq \emptyset)\ [M(x,y) - \{a\} \neq \emptyset].$$

Thus, by applying the attribute a deletion operation, the cardinality of the element $M(x_0, y_0)$ is reduced. Since the number of elements in the matrix and the cardinality of each element are finite, we can produce a matrix M_s containing only the empty set or singleton subsets by a finite number of elementary matrix operations.

For matrix M_s, the union of all its elements satisfies the properties (i) and (ii) of Theorem 2, and hence is a reduct of M_s. According to Lemmas 2 and 3, it is also a reduct of the original discernibility matrix of an information table. By the definition of a minimum matrix, we can conclude that the resulting matrix is in fact a minimum matrix. □

Example 4 *Consider the discernibility matrix of Table 1:*

$$\begin{bmatrix}
 & & & & & \\
\{a,d\} & & & & & \\
\{a,b,c,e\} & \{b,c,d,e\} & & & & \\
\{a,b\} & \{b,d\} & \{c,e\} & & & \\
\{a,c\} & \{a,c,d\} & \{a,b,c,e\} & \{a,b,c\} & & \\
\{d,e\} & \{a,d,e\} & \{a,b,c,d\} & \{a,b,d,e\} & \{a,c,d,e\} & \\
\{b,e\} & \{a,b,d,e\} & \{a,c\} & \{a,e\} & \{a,b,c,e\} & \{b,d\}
\end{bmatrix} \tag{8}$$

We demonstrate matrix simplification by using two different sequences of elementary operations.

(1). By applying the matrix absorption operation on M in Equation (8), the simplified matrix is given by:

$$\begin{bmatrix}
 & & & & & \\
\{a,d\} & & & & & \\
\{a,b\} & \{d,e\} & & & & \\
\{a,b\} & \{b,d\} & \{c,e\} & & & \\
\{a,c\} & \{a,d\} & \{a,b\} & \{a,b\} & & \\
\{d,e\} & \{a,d\} & \{a,b\} & \{a,b\} & \{a,c\} & \\
\{b,e\} & \{a,d\} & \{a,c\} & \{a,e\} & \{a,b\} & \{b,d\}
\end{bmatrix}$$

It can be seen that attribute a can be deleted. The attribute a deletion operation produces:

$$\begin{bmatrix}
 & & & & & \\
\{d\} & & & & & \\
\{b\} & \{d,e\} & & & & \\
\{b\} & \{b,d\} & \{c,e\} & & & \\
\{c\} & \{d\} & \{b\} & \{b\} & & \\
\{d,e\} & \{d\} & \{b\} & \{b\} & \{c\} & \\
\{b,e\} & \{d\} & \{c\} & \{e\} & \{b\} & \{b,d\}
\end{bmatrix}$$

By applying the matrix absorption, we arrive at a minimum matrix:

$$\begin{bmatrix}
 & & & & & & \\
\{d\} & & & & & & \\
\{b\} & \{d\} & & & & & \\
\{b\} & \{b\} & \{c\} & & & & \\
\{c\} & \{d\} & \{b\} & \{b\} & & & \\
\{d\} & \{d\} & \{b\} & \{b\} & \{c\} & & \\
\{b\} & \{d\} & \{c\} & \{e\} & \{b\} & \{b\} &
\end{bmatrix}$$

The corresponding reduct is $\{b, c, d, e\}$.

(2). By applying attribute set $\{b, c\}$ *deletion on* M *in Equation (8), we obtain:*

$$\begin{bmatrix}
 & & & & & & \\
\{a, d\} & & & & & & \\
\{a, e\} & \{d, e\} & & & & & \\
\{a\} & \{d\} & \{e\} & & & & \\
\{a\} & \{a, d\} & \{a, e\} & \{a\} & & & \\
\{d, e\} & \{a, d, e\} & \{a, d\} & \{a, d, e\} & \{a, d, e\} & & \\
\{e\} & \{a, d, e\} & \{a\} & \{a, e\} & \{a, e\} & \{d\} &
\end{bmatrix}$$

We then apply the matrix absorption operation, and the result is a minimum matrix:

$$\begin{bmatrix}
 & & & & & & \\
\{a\} & & & & & & \\
\{a\} & \{d\} & & & & & \\
\{a\} & \{d\} & \{e\} & & & & \\
\{a\} & \{a\} & \{a\} & \{a\} & & & \\
\{d\} & \{a\} & \{a\} & \{a\} & \{a\} & & \\
\{e\} & \{a\} & \{a\} & \{a\} & \{a\} & \{d\} &
\end{bmatrix}$$

We obtain a different reduct $\{a, d, e\}$.

```
Input: The discernibility matrix $M$ of an information table $S$.
Output: A reduct $R$.

while $M$ contains non-singleton subsets do
        $M = \xi(M)$;
output $R$ as the union of all elements in $M$.
```

Fig. 1. A general reduct construction algorithm

4.4 Reduct construction algorithms based on matrix simplification

The matrix operations allow us to design various algorithms based on matrix simplification. By the result of Theorem 3, a high-level outline of a general reduct construction algorithm is given in Figure 1. The operation ξ represents any elementary matrix operation. The sequence of operations is not fixed, and hence many different versions of the algorithm can be designed based on different sequences of operations.

The proposed method for constructing a reduct by matrix simplification shares high-level similarities with the classical method for solving a system of linear equations. Corresponding to the Gaussian elimination method, we propose a row-wise matrix simplification method.

In row-wise matrix simplification, we apply the absorption and the deletion operations to simplify the matrix elements row by row. Consider the situation of simplifying element $M(i,j)$ in a triangular representation of a matrix. As shown by Figure 2, we divide the triangle into two parts. Part $\mathbb{A}$ contains all the simplified elements and part $\mathbb{B}$ contains all the non-simplified elements. If we arrange the elements of the matrix in the following order:

$$M(2,1), M(3,1), M(3,2), M(4,1), \ldots, M(i,j), \ldots, M(n,n-1),$$

where $n = |U|$, then $\mathbb{A}$ consists of those elements before $M(i,j)$ and $\mathbb{B}$ contains those elements after $M(i,j)$.

If element $M(i,j)$ is not the empty set, we will simplify it in three steps. In the first step, we try to absorb $M(i,j)$ by all the elements in $\mathbb{B}$, so that no element in $\mathbb{B}$ is a proper subset of $M(i,j)$. In the second step, we select any attribute a in $M(i,j)$, compute the set of remaining attributes, let $A = M(i,j) - \{a\}$, and replace $M(i,j)$ by $\{a\}$. In the third step, we try to absorb elements in $\mathbb{B}$ by $\{a\}$ and perform the deletion of A on elements in $\mathbb{B}$. It should be noted that after the matrix absorption operation in the first step, for all $\emptyset \neq M(x,y) \in \mathbb{B}$, $M(x,y) - A \neq \emptyset$. The result of this row-wise simplification is a minimum matrix. The detailed algorithm is given in Figure 3.

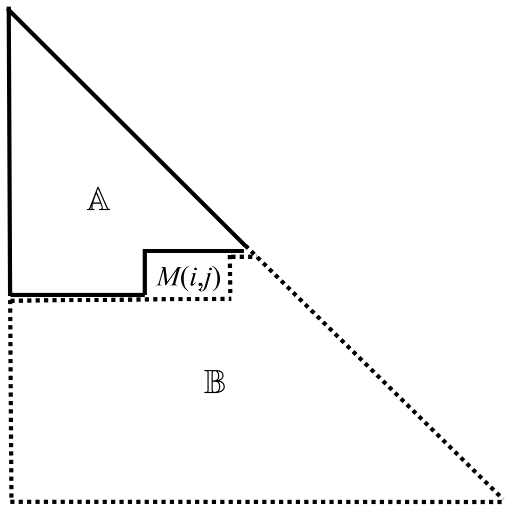

Fig. 2. Divide the triangle of the matrix into two parts: the simplified part ($\mathbb{A}$) and the non-simplified part ($\mathbb{B}$)

There are $\frac{n^2-n}{2}$ elements in the triangular representation of a discernibility matrix. For the row-wise simplification algorithm, each non-empty element $M(i,j)$ is first absorbed by the non-simplified elements in $\mathbb{B}$, then the simplified element $M(i,j)$ is employed to absorb the non-simplified elements in $\mathbb{B}$. If we treat the time complexity of the element absorption operation as a unit, then in the worst cases, element $M(2,1)$ needs to take $\frac{n^2-n}{2}-1$ absorptions, element $M(3,1)$ needs to take $\frac{n^2-n}{2}-2$ absorptions, and so on. Therefore, for all elements in M, the time complexity for the worst case is $O(n^4)$.

In the presentation of the row-wise simplification algorithm, we emphasize its high-level structure. Thus, we purposely omit a number of techniques to speed up the process. For example, if $M(i,j)$ is a singleton set, we do not need to absorb it by elements of $\mathbb{B}$. If $A=\emptyset$, we do not need to perform the deletion of A operation. Normally, many elements of the discernibility matrix are the same set. This suggests a set-based representation. If the matrix is represented as a set, we can eliminate the empty set, and do not need to simplify the duplicate elements [34]. We can select the element with the minimum length as a starting point in order to facilitate the matrix absorption operation.

Example 5 *We illustrate the row-wise simplification algorithm by an example. Suppose we have a decision table shown in Table 3. Its discernibility matrix is shown below, with non-empty elements in part $\mathbb{B}$ underlined:*

```
Input: The discernibility matrix $M$ of an information table $S$.
Output: A reduct $R$.

for $i = 2$ to $n$ do {
    for $j = 1$ to $i - 1$ {
        if $M(i,j) \neq \emptyset$ {
          // Absorb $M(i,j)$ by every non-empty element in $\mathbb{B}$
          for every non-empty element $M(i',j') \in \mathbb{B}$ do
              if $M(i',j') \subset M(i,j)$ then
                  $M(i,j) = M(i',j')$;

          // Divide $M(i,j)$ into two parts
          select an attribute $a$ from $M(i,j)$;
          $A = M(i,j) - \{a\}$;
          $M(i,j) = \{a\}$;

          // Simplify every non-empty element in $\mathbb{B}$
          for every non-empty element $M(i',j') \in \mathbb{B}$ do
              if $a \in M(i',j')$ then
                  $M(i',j') = \{a\}$;
              else
                  $M(i',j') = M(i',j') - A$;
        } // end if
    } // end for loop of $j$
} // end for loop of $i$
```

Fig. 3. A row-wise simplification reduct construction algorithm

Table 3
Another decision table

U	a	b	c	d	e	f	D
1	0	0	0	0	0	0	+
2	1	1	1	1	0	0	+
3	1	0	0	0	1	0	+
4	1	1	0	0	0	1	-
5	1	0	1	0	0	0	-
6	1	0	0	1	0	0	-

$$\begin{bmatrix} & & & & & \\ \emptyset & & & & & \\ \emptyset & \emptyset & & & & \\ \underline{\{a,b,f\}} & \underline{\{c,d,f\}} & \underline{\{b,e,f\}} & & & \\ \underline{\{a,c\}} & \underline{\{b,d\}} & \underline{\{c,e\}} & \emptyset & & \\ \underline{\{a,d\}} & \underline{\{b,c\}} & \underline{\{d,e\}} & \emptyset & \emptyset & \end{bmatrix} \qquad (9)$$

The first iteration of row-wise simplification: absorbing $M(4,1)$ by $\mathbb{B}$ produces:

$$\begin{bmatrix} & & & & \\ \emptyset & & & & \\ \emptyset & \emptyset & & & \\ \{a,b,f\} & \underline{\{c,d,f\}} & \underline{\{b,e,f\}} & & \\ \underline{\{a,c\}} & \underline{\{b,d\}} & \underline{\{c,e\}} & \emptyset & \\ \underline{\{a,d\}} & \underline{\{b,c\}} & \underline{\{d,e\}} & \emptyset & \emptyset \end{bmatrix}$$

Let $A = \{b,f\}$ and $M(4,1) = \{a\}$. We simplify part $\mathbb{B}$ into:

$$\begin{bmatrix} & & & & \\ \emptyset & & & & \\ \emptyset & \emptyset & & & \\ \{a\} & \underline{\{c,d\}} & \underline{\{e\}} & & \\ \underline{\{a\}} & \underline{\{d\}} & \underline{\{c,e\}} & \emptyset & \\ \underline{\{a\}} & \underline{\{c\}} & \underline{\{d,e\}} & \emptyset & \emptyset \end{bmatrix}$$

The second iteration of row-wise simplification: absorbing $M(4,2)$ by $\mathbb{B}$ produces:

$$\begin{bmatrix} & & & & \\ \emptyset & & & & \\ \emptyset & \emptyset & & & \\ \{a\} & \{d\} & \underline{\{e\}} & & \\ \underline{\{a\}} & \underline{\{d\}} & \underline{\{c,e\}} & \emptyset & \\ \underline{\{a\}} & \underline{\{c\}} & \underline{\{d,e\}} & \emptyset & \emptyset \end{bmatrix}$$

Let $A = \emptyset$ and $M(4,2) = \{d\}$. We simplify part $\mathbb{B}$ to produce:

$$\begin{bmatrix} & & & & \\ \emptyset & & & & \\ \emptyset & \emptyset & & & \\ \{a\} & \{d\} & \underline{\{e\}} & & \\ \underline{\{a\}} & \underline{\{d\}} & \underline{\{c,e\}} & \emptyset & \\ \underline{\{a\}} & \underline{\{c\}} & \underline{\{d\}} & \emptyset & \emptyset \end{bmatrix}$$

The third iteration of row-wise simplification: absorbing $M(4,3)$ by $\mathbb{B}$ produces:

$$\begin{bmatrix} & & & & \\ \emptyset & & & & \\ \emptyset & \emptyset & & & \\ \{a\} & \{d\} & \{e\} & & \\ \underline{\{a\}} & \underline{\{d\}} & \underline{\{c,e\}} & \emptyset & \\ \underline{\{a\}} & \underline{\{c\}} & \underline{\{d\}} & \emptyset & \emptyset \end{bmatrix}$$

Let $A = \emptyset$ and $M(4,3) = \{e\}$. We then simplify part $\mathbb{B}$ to produce:

$$\begin{bmatrix} & & & & \\ \emptyset & & & & \\ \emptyset & \emptyset & & & \\ \{a\} & \{d\} & \{e\} & & \\ \underline{\{a\}} & \underline{\{d\}} & \underline{\{e\}} & \emptyset & \\ \underline{\{a\}} & \underline{\{c\}} & \underline{\{d\}} & \emptyset & \emptyset \end{bmatrix}$$

After three iterations, the original matrix is simplified into a minimum matrix, which produces a reduct $\{a, c, d, e\}$.

5 Heuristic Approaches Based on an Ordering of Attributes

Many reduct construction algorithms use an ordering of attributes in order to represent the importance of attributes [7,11,30,33,34,37]. An ordering can be either defined based on a measure of the importance or fitness of attributes, or given by a user [33]. In this section, we demonstrate that such an ordering can be easily incorporated into a matrix simplification process.

5.1 Orderings of attributes

An ordering of attributes can be computed based on a measure of the importance of attributes. The numerical weight of an attribute a may be interpreted as the degree of fitness of a, or the cost of testing a in a rule. Alternatively, an ordering can be given by a user. Formally, for any two attributes $a, b \in At$, $a \succ b$ means that a is more important than b, or equivalently, the user prefers a to b. In the absence of preference, we say that a and b are indifferent, denoted as $a \sim b$. Based on the preference and indifference relations, one can define a relation $\succeq$ on At, that is, $a \succeq b \Longleftrightarrow (a \succ b) \vee (a \sim b)$. It is reasonable to assume that the preference-indifference relation $\succeq$ is reflexive and transitive. A detailed study on this topic in the context of reduct construction is reported by Yao *et al.* [33].

We assume that $\succeq$ on At is transitive and complete (i.e., for $a, b \in At$ either $a \succeq b$ or $b \succeq a$). An ordering satisfying these two properties is called a weak order [5,24]. For a weak order, we can arrange its elements sequentially, called a linear extension [33]. For simplicity, we consider a linear extension of a weak order in the rest of this section.

For a family of attribute sets, we can arrange them sequentially by extending the ordering of attributes. For two sets of attributes, we first arrange attributes in them according to the ordering of attributes. We then compare the two sets of attributes by comparing the corresponding attributes one-by-one. Based on the directions in which attributes are examined, we define two lexical orders [33]. In the left-to-right lexical order, we compare two sets of attributes from left to right in order to determine which set of attributes is preferred. In the right-to-left lexical order, we compare attributes in the reverse order.

Definition 8 (Left-to-right lexical order) *Given two attribute sets $A : a_1 \succeq a_2 \succeq \cdots \succeq a_m$ and $B : b_1 \succeq b_2 \succeq \cdots \succeq b_n$, let $t = \min\{m, n\}$. We say that A precedes B in the left-to-right lexical order, written $A \succ B$, if and only if*

(a) *there exists an i: $1 \le i \le t$ such that $a_j \sim b_j$ for $1 \le j < i$ and $a_i \succ b_i$, or*
(b) *$a_i \sim b_i$ for $1 \le i \le t$ and $m < n$.*

Definition 9 (Right-to-left lexical order) *Given two attribute sets $A : a_1 \succeq a_2 \succeq \cdots \succeq a_m$ and $B : b_1 \succeq b_2 \succeq \cdots \succeq b_n$, let $t = \min\{m, n\}$. We say that A precedes B in the right-to-left lexical order, written $A \succ B$, if and only*

```
Input: The discernibility matrix $M$ of an information table $S$, and an
        ordering $\succeq$ of attributes.
Output: A reduct $R$.

perform matrix absorption on $M$;
while $M$ contains non-singleton subsets do {
    select the right-most attribute $a$ in the ordering $\succeq$ that satisfies
        the condition $\{a\} \notin M$;
    perform the deletion of $a$ on $M$;
    perform matrix absorption on $M$;
} // end while loop
output $R$ as the union of all elements in $M$.
```

Fig. 4. A deletion algorithm for reduct construction

if

(a) *there exists an i:* $0 \leq i < t$ *such that* $a_{m-j} \sim b_{n-j}$ *for* $0 \leq j < i$ *and* $a_{m-i} \succ b_{n-i}$*, or*

(b) $a_{m-i} \sim b_{n-i}$ *for* $0 \leq i < t$ *and* $m < n$*.*

These two lexical orders represent two extreme views and define two different criteria for selecting attribute sets. The left-to-right method focuses on the preferred attributes of the two sets. That is, the winner is determined by comparing the strongest attributes of individual sets. On the other hand, the right-to-left method emphasizes the less preferred attributes of the two sets. The winner is determined by comparing the weakest attributes of individual sets [33].

5.2 Two reduct construction algorithms

For the purpose of constructing a reduct, a given ordering of attributes suggests two strategies. It is reasonable to delete less preferred attributes; it is equally reasonable to include more preferred attributes.

Consider first a deletion-based algorithm that attempts to delete less preferred attributes from right to left according to an ordering of attributes. The algorithm is given in Figure 4. It deletes the least preferred attribute from the matrix in each iteration by the attribute deletion operation until a minimum matrix is constructed.

Example 6 *We illustrate the deletion algorithm by the discernibility matrix of the decision table in Table 3:*

$$\begin{bmatrix} \\ \emptyset \\ \emptyset & \emptyset \\ \{a,b,f\} & \{c,d,f\} & \{b,e,f\} \\ \{a,c\} & \{b,d\} & \{c,e\} & \emptyset \\ \{a,d\} & \{b,c\} & \{d,e\} & \emptyset & \emptyset \end{bmatrix} \tag{10}$$

An ordering of conditional attributes is: $a \succeq b \succeq c \succeq d \succeq e \succeq f$. In this example, the absorbed matrix is equivalent to the one in Equation (10). We first delete the least preferred attribute f from the matrix in Equation (10) and perform matrix absorption to produce:

$$\begin{bmatrix} \\ \emptyset \\ \emptyset & \emptyset \\ \{a,b\} & \{c,d\} & \{b,e\} \\ \{a,c\} & \{b,d\} & \{c,e\} & \emptyset \\ \{a,d\} & \{b,c\} & \{d,e\} & \emptyset & \emptyset \end{bmatrix}.$$

We then delete the second least preferred attribute e from the resulting matrix to produce:

$$\begin{bmatrix} \\ \emptyset \\ \emptyset & \emptyset \\ \{a,b\} & \{c,d\} & \{b,e\} \\ \{a,c\} & \{b,d\} & \{c,e\} & \emptyset \\ \{a,d\} & \{b,c\} & \{d,e\} & \emptyset & \emptyset \end{bmatrix}$$

and perform the matrix absorption to produce:

```
Input: The discernibility matrix $M$ of an information table $S$, and an
        ordering $\succeq$ of attributes.
Output: A reduct $R$.

perform matrix absorption on $M$;
while $M$ contains non-singleton subsets do {
    select the left-most attribute $a$ in the ordering $\succeq$ that satisfies
        the condition $\{a\} \notin M$;
    select an element $M(x_0, y_0)$ that satisfies the condition $a \in M(x_0, y_0)$;
    $A = M(x_0, y_0) - \{a\}$ and $M(x_0, y_0) = \{a\}$;
    perform the deletion of $A$ on $M$;
    perform matrix absorption on $M$;
} // end while loop
output $R$ as the union of all elements in $M$.
```

Fig. 5. An addition algorithm for reduct construction

$$\begin{bmatrix} & & & & & \\ \emptyset & & & & & \\ \emptyset & \emptyset & & & & \\ \{b\} & \{c\} & \{b\} & & & \\ \{c\} & \{b\} & \{c\} & \emptyset & & \\ \{d\} & \{b\} & \{d\} & \emptyset & \emptyset & \end{bmatrix}$$

We obtain a minimum matrix after two deletion operations. The deletion algorithm results in the winning reduct $\{b, c, d\}$ under the right-to-left lexical order.

For the deletion algorithm, there is no guarantee that more preferred attributes will be in the constructed reduct. In order to resolve this problem, we consider an addition-based algorithm that tries to add more preferred attributes to a partial reduct from left to right according to the ordering of attributes. The algorithm is given in Figure 5.

Given an attribute $a \in M(x, y)$, the attribute set $A = M(x, y) - \{a\}$ is called the tail of a in $M(x, y)$. In the addition-based algorithm, if we want to add a to a partial reduct, we need to delete the tail of a, so that a would be necessary to discern the corresponding pair of objects. We consider two different situations. If there is only one element $M_0(x, y)$ containing a and it is not a singleton set, then the only tail $A = M(x_0, y_0) - \{a\}$ is deleted from all elements of M. If there exist more than one element containing a, then a has more than one tail. We can compare the tails based on either the left-to-right or the right-to-left lexical order, and select the least preferred one for tail deletion. The reason to delete the least preferred tail is try to keep the most preferred attributes in the partial reduct. It should be noted that after the matrix absorption operation, $\forall M(x, y) \neq \emptyset\ [(M(x, y) - A) \neq \emptyset]$. Thus, one can safely perform the deletion of A operation.

While the deletion algorithm can generate the winning reduct under the right-to-left lexical order, the addition algorithm does not necessarily produce the winning reduct under the left-to-right lexical order. In fact, constructing the winning reduct under the left-to-right lexical order has been proven to be NP-hard [13]. The intuitive explanation is that the attribute set deletion operation combined with the matrix absorption operation may delete attributes of the winning reduct under the left-to-right lexical order.

Example 7 *We illustrate the addition algorithm by the same matrix in Equation (10) of the decision table in Table 3. The ordering of conditional attributes is: $a \succeq b \succeq c \succeq d \succeq e \succeq f$. In order to keep the most preferred attribute a in the minimum matrix, we can either delete the tail $\{d\}$ according to the left-to-right lexical order, or delete the tail $\{b, f\}$ according to the right-to-left lexical order.*

We first show the process of the addition algorithm by selecting the least preferred tail under the left-to-right lexical order. By selecting a, we delete the least preferred tail $\{d\}$ under the left-to-right order from M in Equation (10), and then perform matrix absorption:

$$\begin{bmatrix} & & & & & \\ \emptyset & & & & & \\ \emptyset & \emptyset & & & & \\ \{a\} & \{c, f\} & \{b\} & & & \\ \{a\} & \{b\} & \{e\} & \emptyset & & \\ \{a\} & \{b\} & \{e\} & \emptyset & \emptyset & \end{bmatrix}$$

We then select c, delete $\{f\}$ from the resulting matrix, and then perform matrix absorption:

$$\begin{bmatrix} & & & & & \\ \emptyset & & & & & \\ \emptyset & \emptyset & & & & \\ \{a\} & \{c\} & \{b\} & & & \\ \{a\} & \{b\} & \{e\} & \emptyset & & \\ \{a\} & \{b\} & \{e\} & \emptyset & \emptyset & \end{bmatrix}$$

The resulting matrix is a minimum matrix, which indicates a reduct $\{a, b, c, e\}$.

We then show the process of the addition algorithm by selecting the least preferred tail under the right-to-left lexical order. By selecting a*, we delete the least preferred tail* $\{b, f\}$ *under the right-to-left order from* M *in Equation (10), and then perform matrix absorption:*

$$\begin{bmatrix} & & & & & \\ \emptyset & & & & & \\ \emptyset & \emptyset & & & & \\ \{a\} & \{c\} & \{e\} & & & \\ \{a\} & \{d\} & \{e\} & \emptyset & & \\ \{a\} & \{c\} & \{d\} & \emptyset & \emptyset & \end{bmatrix}$$

The resulting matrix is a minimum matrix, which indicates a reduct $\{a, c, d, e\}$*. It is easy to identify that two resulting reducts satisfying* $\{a, b, c, e\} \succ \{a, c, d, e\}$ *according to the left-to-right lexical order. Keeping the current most preferred attribute by deleting its least preferred tail cannot guarantee that the next most preferred attribute can be kept. It demonstrates that the addition algorithm does not necessarily produce the winning reduct under the left-to-right lexical order.*

6 Conclusion

The study of reducts is fundamental in rough set theory. The concept of a discernibility matrix enables us to establish a logical and theoretical foundation for reducts of an information table. One can define a Boolean function based on a discernibility matrix. By applying the absorption and distribution laws, the discernibility function is transformed to a reduced disjunctive form. Each prime implicant corresponds to a reduct of the information table. Although many reduct construction algorithms have been proposed based on the discernibility matrix, they are based on the logical interpretations of the matrix, rather than processing the matrix directly. On the other hand, directly operating on the matrix may provide us an alternative view and produce a deeper understanding of reduct construction.

Based on the notion of a minimum matrix suggested by Pawlak and a reduct construction method proposed by Wang and his colleagues, we introduce a method of reduct construction by matrix simplification. Specifically, one directly operates on the matrix, in a similar way to the classical Gaussian elimination method for solving a system of linear equations. The basic notions of this method are matrix equivalence and matrix specification. For transforming a matrix to an equivalent matrix, we introduce element and matrix absorptions. For transforming a matrix to one of its specifications, we introduce single attribute and attribute set deletions. These operations are sufficient to produce a minimum matrix in a finite number of steps. By the fact that the union of all elements in a minimum matrix is a reduct, the matrix simplification process in fact offers a procedure for constructing a reduct.

To demonstrate the working principle of the method, we outline a high-level algorithm and give a concrete row-wise simplification algorithm. The method of matrix simplification is applied to the situations where we have an ordering of attributes. Two algorithms are proposed by exploring the orderings of attributes. A deletion-based algorithm tries to delete unimportant attributes, and an addition-based algorithm tries to include important attributes.

For clarity, we did not cover any topics on optimizations related to the implementation of the proposed matrix simplification algorithms. This allows us to focus on the basic issues of the method in order to obtain new insights into reduct construction. As future work, we will investigate data structures for representing a discernibility matrix which will produce more efficient algorithms.

Acknowledgement

The authors thank the anonymous referees for the constructive comments and suggestions. This work is partially supported by a Discovery Grant from NSERC Canada.

References

[1] J.G. Bazan, H.S. Nguyen, S.H. Nguyen, P. Synak, J. Wroblewski, Rough set algorithms in classification problem, In: L. Polkowski, S. Tsumoto, T.Y. Lin (Eds.), Rough Set Methods and Applications, 2000, pp. 49-88.

[2] T. Beaubouef, F.E. Petry, G. Arora, Information-theoretic measures of uncertainty for rough sets and rough relational databases, Information Sciences 109 (1998) 185-195.

[3] R.B. Bhatt, M. Gopal, On the compact computational domain of fuzzy-rough sets, Pattern Recognition Letters 6 (2005) 1632-1640.

[4] D.G. Chen, C.Z. Wang, Q.H. Hu, A new approach to attribute reduction of consistent and inconsistent covering decision systems with covering rough sets, Information Sciences 177 (2007) 3500-3518.

[5] P.C. Fishburn, Utility Theory for Decision-Making, John Wiley & Sons, New York, 1970.

[6] J.W. Guan, D.A. Bell, Rough computational methods for information systems, Artificial Intelligence 105 (1998) 77-103.

[7] S.Q. Han, J. Wang, Reduct and attribute order, Journal of Computer Science and Technology 19 (2004) 429-449.

[8] F. Hu, G.Y. Wang, Quick reduction algorithm based on attribute order, Chinese Journal of Computers 30 (2007) 1429-1435.

[9] Q.H. Hu, Z.X. Xie, D.R. Yu, Hybrid attribute reduction based on a novel fuzzy-rough model and information granulation, Pattern Recognition 40 (2007) 3509-3521.

[10] Q.H. Hu, D.R. Yu, Z.X. Xie, Information-preserving hybrid data reduction based on fuzzy-rough techniques, Pattern Recognition Letters 27 (2006) 414-423.

[11] R. Jensen, Q. Shen, Semantics-preserving dimensionality reduction: Rough and fuzzy-rough-based approaches, IEEE Transactions on Knowledge and Data Engineering 16 (2004) 1457-1471.

[12] Y. Leung, W.Z. Wu, W.X. Zhang, Knowledge acquisition in incomplete information systems: A rough set approach, European Journal of Operational Research 168 (2006) 164-180.

[13] H.L. Liang, J. Wang, Y.Y. Yao, User-oriented feature selection for machine learning, Computer Journal 50 (2007) 421-434.

[14] J.Y. Liang, Z.Z. Shi, The information entropy, rough entropy and knowledge granulation in rough set theory, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 12 (2004) 3746.

[15] J.S. Mi, W.Z. Wu, W.X. Zhang, Approaches to knowledge reduction based on variable precision rough set model, Information Sciences 159 (2004) 255-272.

[16] H.S. Nguyen, On the decision table with maximal number of reducts, Electronic Notes in Theoretical Computer Science 82 (2003) 198-205.

[17] H.S. Nguyen, Approximate Boolean reasoning: Foundations and applications in data mining, Transactions on Rough Sets 5 (2006) 334-506 LNCS 4100.

[18] S.H. Nguyen, H.S. Nguyen, Some efficient algorithms for rough set methods, Proceedings of the International Conference on Information Processing and Management of Uncertainty on Knowledge Based Systems, 1996, pp. 1451 1456.

[19] Z. Pawlak, Rough sets, International Journal of Computer Information and Science 11 (1982) 341-356.

[20] Z. Pawlak, Rough Sets: Theoretical Aspects of Reasoning About Data, Kluwer Academic Publishers, Dordrecht, MA, 1991.

[21] Z. Pawlak, A. Skowron, Rudiments of rough sets, Information Sciences 177 (2007) 3-27.

[22] Z. Pawlak, A. Skowron, Rough sets and Boolean reasoning, Information Sciences 177 (2007) 41-73.

[23] M. Quafafou, α-RST: A generalization of rough set theory, Information Sciences 124 (2000) 301-316.

[24] F.S. Roberts, Measurement Theory, Academic Press, New York, 1976.

[25] RSES http://alfa.mimuw.edu.pl/∼rses

[26] A. Skowron, C. Rauszer, The discernibility matrices and functions in information systems, In: R. Slowiński (Ed.), Intelligent Decision Support, Handbook of Applications and Advances of the Rough Sets Theory, Dordrecht, Kluwer, 1992.

[27] R.W. Swiniarski, Rough sets methods in feature reduction and classification, International Journal of Applied Mathematics and Computer Science 11 (2001) 565-582.

[28] G.Y. Wang, H. Yu, D. Yang, Decision table reduction based on conditional information entropy, Chinese Journal of Computers 25 (2002) 759-766.

[29] J. Wang, D.Q. Miao, Analysis on attribute reduction strategies of rough set, Chinese Journal of Computer Science and Technology 13 (1998) 189-192.

[30] J. Wang, J. Wang, Reduction algorithms based on discernibility matrix: the ordered attributes method, Journal of Computer Science and Technology 16 (2001) 489-504.

[31] S.K.M. Wong, W. Ziarko, On optimal decision rules in decision tables, Bulletin of Polish Academy of Sciences 33 (1985) 693-696.

[32] W.Z. Wu, M. Zhang, H.Z. Li, J.S. Mi, Knowledge reduction in random information systems via Dempster-Shafer theory of evidence, Information Sciences 174 (2005) 143-164.

[33] Y.Y. Yao, Y. Zhao, J. Wang, S.Q. Han, A model of machine learning based on user preference of attributes, Proceedings of International Conference on Rough Sets and Current Trends in Computing, 2006, pp. 587-596.

[34] K. Zhao, J. Wang, A reduction algorithm meeting users' requirements, Journal of Computer Science and Technology 17 (2002) 578-593.

[35] W. Zhu, F.Y. Wang, Reduction and axiomization of covering generalized rough sets, Information Sciences 152 (2003) 217-230.

[36] W. Zhu, F.Y. Wang, On three types of covering-based rough sets, IEEE Transactions on Knowledge and Data Engineering 19 (2007) 1131-1144.

[37] W. Ziarko, Rough set approaches for discovering rules and attribute dependencies, In: W. Klösgen, J.M. Żytkow (Eds.), Handbook of Data Mining and Knowledge Discovery, Oxford, 2000, pp. 328-339.