

Article

# A Quick Algorithm for Binary Discernibility Matrix Simplification using Deterministic Finite Automata

Nan Zhang <sup>1,\*</sup>, Baizhen Li <sup>1,2</sup>, Zhongxi Zhang <sup>1</sup> and Yanyan Guo <sup>1</sup>

- <sup>1</sup> School of Computer and Control Engineering, Yantai University, 264005 Yantai, China; 1833009@tongji.edu.cn (B.L.); zhangzhongxi89@gmail.com (Z.Z.); smallgyy@ytu.edu.cn (Y.G.)
- <sup>2</sup> Department of Computer Science and Technology, Tongji University, 201804 Shanghai, China

\* Correspondence: zhangnan@ytu.edu.cn; Tel.: +86-186-1535-1131

Received: 30 October 2018; Accepted: 6 December 2018; Published: 7 December 2018



**Abstract:** The binary discernibility matrix, originally introduced by Felix and Ushio, is a binary matrix representation for storing discernible attributes that can distinguish different objects in decision systems. It is an effective approach for feature selection, knowledge representation and uncertainty reasoning. An original binary discernibility matrix usually contains redundant objects and attributes. These redundant objects and attributes may deteriorate the performance of feature selection and knowledge acquisition. To overcome this shortcoming, row relations and column relations in a binary discernibility matrix are defined in this paper. To compare the relationships of different rows (columns) quickly, we construct deterministic finite automata for a binary discernibility matrix. On this basis, a quick algorithm for binary discernibility matrix simplification using deterministic finite automata (BDMSDFA) is proposed. We make a comparison of BDMR (an algorithm of binary discernibility matrix reduction), IBDMR (an improved algorithm of binary discernibility matrix reduction) and BDMSDFA. Finally, theoretical analyses and experimental results indicate that the algorithm of BDMSDFA is effective and efficient.

Keywords: rough sets; binary discernibility matrices; deterministic finite automata

# 1. Introduction

Decision making can be considered as the process of choosing the best alternative from the feasible alternatives. With the development of research, decision making is extended from one attribute to multiple attributes. To solve problems in multiple attribute decision making, various theories such as fuzzy sets, rough sets and utility theory, etc. have been used. Many of significant results [1–8] have been achieved in multiple attribute decision making. Researchers in rough set theory [9] are usually concerned with attribute reduction (or feature selection) problems of multiple attribute decision making. The binary discernibility matrix, proposed by Felix and Ushio [10], is a useful tool for attribute reduction and knowledge acquisition. Recently, many algorithms of attribute reduction based on binary discernibility matrices have been developed [11–13]. In 2014, Zhang et al. [14] proposed a binary discernibility matrix for an incomplete information system, and designed a novel algorithm of attribute reduction based on the proposed binary discernibility matrix. In the paper [15], Li et al. developed an attribute reduction algorithm in terms of the improved binary discernibility matrix, and applied the algorithm in customer relationship management. Tiwari et al. [16] developed hardware for a binary discernibility matrix which can be used for attribute reduction and rule acquisition in an information system. Considering mathematical properties of a binary discernibility matrix, Zhi and Miao [17] introduced the so-called binary discernibility matrix reduction (BDMR), which was actually an algorithm for binary discernibility matrix simplification. On the basis of BDMR, two algorithms for attribute reduction and reduction judgement were presented.



A binary discernibility matrix with a vertical partition [18] was proposed to deal with big data in attribute reduction. Ren et al. [19] constructed an improved binary discernibility matrix which can be used in an inconsistent information system. Ding et al. [20] discussed several problems about a binary discernibility matrix in an incomplete system. Combining the binary discernibility matrix in an incomplete system. Combining the binary discernibility matrix in an incomplete system, an algorithm of incremental attribute reduction was proposed. In the paper [20], a novel method for calculation of incremental core attribute was introduced firstly. On this basis, an algorithm of attribute reduction was proposed. As is well known that core attributes play a crucial role in heuristic attribute reduction algorithms. Core attributes are computationally expensive in attribute reduction. Hu et al. [21] gave a quick algorithm of the core attribute calculation using a binary discernibility matrix. The computational complexity of the algorithm is O(|C||U|), where |C| is the number of condition attributes and |U| is the number of objects in the universe.

An original binary discernibility matrix usually contains redundant objects and attributes. These redundant objects and attributes may deteriorate the performance of feature selection (attribute reduction) and knowledge acquisition based on binary discernibility matrices. In other words, storing or processing all objects and attributes in an original binary discernibility matrix could be computationally expensive, especially in dealing with large scale data sets with high dimensions. So far, however, few works about the binary discernibility matrix simplification have been investigated. The existing algorithms regarding binary discernibility matrix simplification are time-consuming. To tackle this problem, our works in this paper concern on how to improve the time efficiency of algorithms of binary discernibility matrix simplification. On this purpose, we construct deterministic finite automata in a binary discernibility matrix to compare the relationships of different rows (or columns) quickly. By using deterministic finite automata, we develop a quick algorithm of binary discernibility matrix simplification. Experimental results show that the proposed algorithm is effective and efficient. The contributions of this paper are summarized as follows: First, we define row and column relations which can be used for constructing deterministic finite automata in a binary discernibility matrix. Second, deterministic finite automata in a binary discernibility matrix are proposed to compare the relationships of different rows (or columns) quickly. Third, based on this method, a quick algorithm for binary discernibility matrix simplification (BDMSDFA) is proposed. The proposed method in this paper is meaningful in practical applications. First, by using BDMSDFA, we obtain the simplified binary discernibility matrices quickly. These simplified binary discernibility matrices can significantly improve the efficiency of attribute reduction (feature selection) in decision systems. Second, a binary discernibility matrix without redundant objects and attributes will have the high performance of learning algorithms, and need less space for data storage.

The rest of this paper is structured as follows. We review basic notions about rough set theory in the next section. In Section 3, we propose a general binary discernibility matrix, and define row relations and column relations in a binary discernibility matrix. In Section 4, we develop a quick algorithm for binary discernibility matrix simplification which is called BDMSDFA. Experimental results in Section 5 show that the algorithm of BDMSDFA is effective and efficient, it can be applicable to simplification of large-scale binary discernibility matrices. Finally, the whole paper is summarized in Section 6.

#### 2. Preliminaries

Basic notions about rough set theory are briefly reviewed in this section. Some further details about rough set theory can be found in the paper [9]. A Pawlak decision system can be regarded as an original information system with decision attributes which give decision classes for objects.

A Pawlak decision system [9] can be denoted by 4-tuple DS = (U, AT, V, f), where universe  $U = \{x_1, x_2, ..., x_{|U|}\}$  is a finite non-empty set of objects; attribute set  $AT = C \cup D$ ,  $C \cap D = \emptyset$ , where  $C = \{a_1, a_2, ..., a_{|C|}\}$  is called a condition attribute set and  $D = \{d\}$  is called a decision attribute set in a decision system;  $V_{a_m}$  is the domain of a condition attribute  $a_m \in AT$ ,  $V = \bigcup_{a_m \in AT} V_{a_m}$  and

 $f : U \times AT \rightarrow V$  is a function such that  $f(x_i, a_m) = a_m(x_i) \in V_{a_m \in AT}$ ,  $f(x_i, d) = d(x_i) \in V_{d \in AT}$ , where  $x_i \in U$ .

Given a Pawlak decision system  $DS = (U, C \cup D, V, f)$ , for  $\forall x_i, x_j \in U$ , an indiscernibility relation regarding attribute set  $B \subseteq C$  is defined as  $IND(B) = \{(x_i, x_j) : \forall b \in B, f(x_i, b) = f(x_j, b)\}$ . Therefore, the discernibility relation regarding attribute set  $B \subseteq C$  is given by  $DIS(B) = \{(x_i, x_j) : \exists b \in B, f(x_i, b) \neq f(x_j, b)\}$ . The indiscernibility relation regarding  $B \subseteq C$  is reflexive, symmetric and transitive. Meanwhile, the discernibility relation is irreflexive, symmetric, but not transitive. A partition of U derived from IND(B) is denoted by U/IND(B). The equivalence class in U/IND(B)containing object  $x_i$  is defined as  $[x_i]_{IND(B)} = [x_i]_B = \{x_i \in U : (x_i, x_i) \in IND(B)\}$ .

For  $\forall B \subseteq C$ , the relative indiscernibility relation and discernibility relation with respect to decision attribute set [9] are defined by:

$$IND(B|D) = \{(x_i, x_j) : x_i, x_j \in U, (\forall b \in B \to (f(x_i, b) = f(x_j, b)) \lor (f(x_i, d) = f(x_j, d))\},\$$
$$DIS(B|D) = \{(x_i, x_j) : x_i, x_j \in U, (\exists b \in B \to (f(x_i, b) \neq f(x_j, b)) \land (f(x_i, d) \neq f(x_j, d))\}.$$

A relative indiscernibility relation IND(B|D) with respect to  $B \subseteq C$  is reflexive, symmetric, but not transitive. A relative discernibility relation DIS(B|D) with respect to  $B \subseteq C$  is irreflexive, symmetric, but not transitive.

A discernibility matrix, proposed by Skowron and Rauszer [22], suggests a matrix representation for storing condition attribute sets which can discern objects in the universe. Discernibility matrix is an effective method in reduct construction, data representation and rough logic reasoning, and it is also useful mathematical tool in data mining, machine learning, etc. Many extended models of dicernibility matrices have been studied in recent years [23–30]. Considering the classification property  $\Delta$ , Miao et al. [31] constructed a general discernibility matrix  $M_{\Delta} = (m_{\Delta}(x_i, x_j))$ , where  $m_{\Delta}(x_i, x_j)$  is denoted by:

$$m_{\Delta}(x_i, x_j) = \begin{cases} \{a \in C : f(x_i, a) \neq f(x_j, a)\}, & (x_i, x_j) \in DIS_{\Delta}(C|D) \\ \emptyset & \text{otherwise} \end{cases}$$

where  $(x_i, x_j) \in DIS_{\Delta}(C|D)$  denotes objects  $x_i$  and  $x_j$  are discernible with respect to the classification property  $\Delta$  in a decision system *DS*. It should be noted that  $\Delta$  is a general definition on classification property. A general discernibility matrix provides a common solution to attribute reduction algorithms based on discernibility matrices. By constructing different discernibility matrices, the relative attribute reducts with different reduction targets can be obtained. Based on the relative discernibility relation DIS(C|D), Miao et al. [31] introduced a relationship preservation discernibility matrix which can be denoted as follows:

**Definition 1.** [31] Let  $DS = (U, C \cup D, V, f)$  be a decision system, for  $\forall x_i, x_j \in U, \forall a \in C, 1 \leq i < j \leq |U|, M_{relationship} = (m_{relationship}(x_i, x_j))$  is a relationship preservation discernibility matrix, where  $m_{relationship}(x_i, x_j)$  is defined by:

$$m_{relationship}(x_i, x_j) = \begin{cases} \{a \in C : f(x_i, a) \neq f(x_j, a)\} & (x_i, x_j) \in DIS(C|D) \\ \emptyset & otherwise \end{cases}$$

#### 3. Binary Discernibility Matrices and Their Simplifications

The binary discernibility matrix, initiated by Felix and Ushio [10], is a binary presentation of original discernibility matrix. In this section, we suggest a general binary discernibility matrix. Relations of row pairs and column pairs are discussed respectively. Formally, a binary discernibility matrix [10] is introduced as follows:

**Definition 2.** [10] Given a decision system  $DS = (U, C \cup D, V, f)$ , for  $\forall x_i, x_j \in U$  and  $\forall a_m \in C$ .  $M_{BDM} = (m_{BDM}(x_i, x_j))$  is a binary discernibility matrix, where the element  $m_{BDM}(x_i, x_j)$  is denoted by:

$$m_{BDM}(x_i, x_j) = \begin{cases} 1 & f(x_i, a_m) \neq f(x_j, a_m) \land d(x_i) \neq d(x_j) \\ 0 & otherwise \end{cases}$$

Based on a binary discernibility matrix, discernible attributes about  $x_i$  and  $x_j$  can be easily obtained. A binary discernibility matrix brings us an understandable approach for representations of discernible attributes, and can be used for designing reduction algorithms. To satisfy more application requirements, we extend original binary discernibility matrix to general binary discernibility matrix as follows:

**Definition 3.** Given a decision system  $DS = (U, C \cup D, V, f)$ , for  $\forall x_i, x_j \in U$ ,  $\forall a_m \in C$ ,  $M_{BDM}^{\Delta} = (m_{BDM}^{\Delta}(x_i, x_j))$  regarding  $\Delta$  is a general binary discernibility matrix, in which  $m_{BDM}^{\Delta}(x_i, x_j)$  is defined by:

$$m^{\Delta}_{BDM}(x_i, x_j) = \begin{cases} 1 & (x_i, x_j) \in DIS_{\Delta}(C|D) \\ 0 & otherwise \end{cases}$$

 $DIS_{\Delta}(C|D)$  is the discernibility relation regarding classification property  $\Delta$ . The set of rows in  $M_{BDM}^{\Delta}$ is presented by  $R = \{r_1, r_2, ..., r_{|R|}\}$ , where  $|R| = (|U| \times (|U| - 1))/2$ . The set of columns in  $M_{BDM}^{\Delta}$ is presented by  $C = \{a_1, a_2, ..., a_{|C|}\}$ , where |C| is the cardinality of attribute sets in a decision system. For convenience, a general binary discernibility matrix  $M_{BDM}^{\Delta} = (m_{BDM}^{\Delta}(x_i, x_j))$  can be also denoted by  $M_{BDM}^{\Delta} = (m_{BDM}^{\Delta}(e_p^m, e_q^m))$ . For  $\forall a_m \in C, \forall r_p, r_q \in R, e_p^m$  is the matrix element at row  $r_p$  and column  $a_m$  in  $M_{BDM}^{\Delta}$ , and  $e_q^m$  is the matrix element at row  $r_q$  and column  $a_m$  in  $M_{BDM}^{\Delta}$ , where  $1 \leq p < q \leq$  $(|U| \times (|U| - 1))/2, 1 \leq m \leq |C|$ .

Since a general binary discernibility matrix provides a common structure of binary discernibility matrices in rough set theory, one can construct a binary discernibility matrix according to a given classification property. Any binary discernibility matrix can be also regarded as the special case of the general binary discernibility matrix. Therefore, a general definition of binary discernibility matrix is necessary and important. Based on the relative discernibility relation with respect to *D*, Definition 2 can be also rewritten as follows:

**Definition 4.** Given a decision system  $DS = (U, C \cup D, V, f)$ , for  $\forall x_i, x_j \in U, \forall a_m \in C, DIS(C|D)$  is the relative discernibility relation with respect to a condition attribute set C.  $M_{BDM} = (m_{BDM}(x_i, x_j))$  is a binary discernibility matrix, in which the element  $m_{BDM}(x_i, x_j)$  is denoted by:

$$m_{BDM}(x_i, x_j) = \begin{cases} 1 & (x_i, x_j) \in DIS(C|D) \\ 0 & otherwise \end{cases}$$

This definition is equivalent to Definition 2 [10]. It is noted that we calculate binary discernibility matrix in this paper by using the relationship preservation discernibility matrix.

**Definition 5.** For  $\forall a_m \in C$ ,  $\forall r_p, r_q \in R$ ,  $e_p^m$  and  $e_q^m$  are elements in a binary discernibility matrix  $M_{BDM} = (m_{BDM}(x_i, x_j))$ , a row pair with respect to attribute  $a_m$  is denoted by  $\langle e_p^m, e_q^m \rangle \in \{\langle 0, 0 \rangle, \langle 0, 1 \rangle, \langle 1, 0 \rangle, \langle 1, 1 \rangle\}$ , a binary relation between row  $r_p$  and  $r_q$  is defined as  $R_{row} = \{\langle e_p^m, e_q^m \rangle : 1 \leq p < q \leq |R|\}$ .

Similar to Definition 5, we define a column pair and a binary relation with respect to columns as follows.

**Definition 6.** For  $\forall a_m, a_n \in C$ ,  $\forall r_p \in R$ , elements  $e_p^m$  and  $e_p^n$  in a binary discernibility matrix  $M_{BDM} =$  $(m_{BDM}(x_i, x_j))$ , a column pair with respect to row  $r_p$  is denoted by  $\langle e_p^m, e_p^n \rangle \in \{\langle 0, 0 \rangle, \langle 0, 1 \rangle, \langle 1, 0 \rangle, \langle 0, 1 \rangle, \langle 1, 0 \rangle\}$ <1,1>}, a binary relation between column  $a_m$  and  $a_n$  is defined as  $R_{col} = \{< e_p^m, e_p^n >: 1 \le m < n \le |C|\}$ .

For the matrix element  $e_p^m$  and  $e_q^m$  in the same column  $a_m$ , we define three row relations in a binary discernibility matrix as follows.

**Definition 7.** Given a binary discernibility matrix  $M_{BDM} = (m_{BDM}(x_i, x_j)), \forall r_p, r_q \in R$ ,

- for  $\forall a_m \in C$ ,  $\exists a_n \in C$ ,  $r_p \supset r_q$  if and only if  $e_p^m + e_q^m = e_p^m$  and  $e_p^n \neq e_q^n$ ; for  $\forall a_m \in C$ ,  $\exists a_n \in C$ , (1)
- (2)

 $r_q \supset r_p$  if and only if  $e_q^m + e_p^m = e_q^m$  and  $e_q^n \neq e_p^n$ ; for  $\forall a_m \in C$ ,  $r_p = r_q$  if and only if  $e_p^m = e_q^m$ ; for  $\exists a_m, a_n \in C$ ,  $r_p \neq r_q$  if and only if  $e_p^m + e_q^m = e_q^m (e_p^m \neq e_q^m)$  and  $e_p^n + e_q^n = e_p^n (e_p^n \neq e_q^n)$ . (3)

Analogous to Definition 7, for matrix elements  $e_p^m$  and  $e_p^n$  in the same row  $r_p$ , we define column relations in a binary discernibility matrix as follows.

**Definition 8.** Given a binary discernibility matrix  $M_{BDM} = (m_{BDM}(x_i, x_j)), \forall a_m, a_n \in C$ ,

- for  $\forall r_p \in R$ ,  $\exists r_q \in R$ ,  $a_m \supset a_n$  if and only if  $e_p^m + e_p^n = e_p^m$  and  $e_q^m \neq e_q^n$ ; for  $\forall r_p \in R$ ,  $\exists r_q \in R$ , (1) $a_n \supset a_m$  if and only if  $e_p^n + e_p^m = e_p^n$  and  $e_q^n \neq e_q^m$ ; for  $\forall r_p \in R$ ,  $a_m = a_n$  if and only if  $e_p^m = e_p^n$ ;
- (2)

(3) for 
$$\exists r_p, r_q \in \mathbb{R}$$
,  $a_m \neq a_n$  if and only if  $e_p^m + e_p^n = e_p^m (e_p^m \neq e_p^n)$  and  $e_q^m + e_q^n = e_q^n (e_q^m \neq e_q^n)$ .

Let  $A_p$  be the elements' set of a prime implicant in a disjunctive normal form with row  $r_p$  and  $A_q$  be the elements' set of a prime implicant in a disjunctive normal form with row  $r_q$ , then  $r_p \supset r_q$ means that  $A_p$  is the superset of  $A_q$ . For  $\forall a_m, a_n \in C$ ,  $a_m \supset a_n$  indicates attribute  $a_m$  can distinguish more objects in the universe. In a binary discernibility matrix, the row in which all elements are 0s indicates there are no attribute can discern the related objects, and the column in which all elements are 0s indicates that this attribute cannot discern objects in the universe.

In [17], Zhi and Miao first proposed an algorithm of a binary discernibility matrix simplification shown in Algorithm 1. To improve the efficiency of BDMR, Wang et al. [32] introduced an improved algorithm of binary discernibility matrix reduction shown in Algorithm 2.

#### Algorithm 1 : An algorithm of binary discernibility matrix reduction, BDMR.

```
Input: Original binary discernibility matrix M<sub>BDM</sub>;
Output: Simplified binary discernibility matrix M<sub>BDM</sub>
 1: delete the row in which all elements are 0 s;
 2: for p = 1 to |R| do
 3:
        for q = 1 to |R| do
            if r_p \supset r_q then
 4:
               delete row r_p
 5:
 6:
               break
 7:
            end if
 8:
        end for
 9: end for
10: delete the column in which all elements are 0 s;
11: for m = 1 to |C| do
12:
        for n = 1 to |C| do
13:
            if a_n \supset a_m then
               delete column am
14:
15:
               break
            end if
16:
        end for
17:
18: end for
19: output a simplified binary discernibility matrix M'_{BDM};
```

#### Algorithm 2 : An improved algorithm of binary discernibility matrix reduction, IBDMR.

**Input:** Original binary discernibility matrix *M*<sub>BDM</sub>; **Output:** Simplified binary discernibility matrix  $M_{BDM}$ 1: delete the row in which all elements are 0 s; 2: sort rows in ascending order by the quantity of the number '1' in each row for p = 1 to |R| do 3: for q = 1 to |R| do 4: if  $r_p \supset r_q$  then 5: 6: delete row  $r_p$ 7: break 8: end if end for 9: 10: end for 11: delete the column in which all elements are 0 s; 12: **for** m = 1 **to** |C| **do** for n = 1 to |C| do 13: 14: if  $a_n \supset a_m$  then delete column *a<sub>m</sub>* 15: break 16: 17: end if end for 18: 19: end for 20: output a simplified binary discernibility matrix  $M'_{BDM}$ ;

#### 4. A Quick Algorithm for Binary Discernibility Matrix Simplification

In this section, we investigate two theorems related to row relations and column relations respectively. Based on the two theorems, deterministic finite automata for row and column relations are introduced. Deterministic finite automata can be carried out to obtain the row relations and column relations quickly. By using deterministic finite automata, we propose an algorithm of binary discernibility matrix simplification using deterministic finite automata (BDMSDFA).

**Theorem 1.** Let  $M_{BDM} = (m_{BDM}(x_i, x_j))$  be a binary discernibility matrix, for  $\forall a_m \in C, \forall r_p, r_q \in R$ ,  $|R| = (|U| \times (|U| - 1))/2$ , we have:

- (1)
- *if*  $r_p \supset r_q$ , then there exists  $< 1, 0 > \in \{< e_p^m, e_q^m >: 1 \le p < q \le |R|\};$ *if*  $r_q \supset r_p$ , then there exists  $< 0, 1 > \in \{< e_p^m, e_q^m >: 1 \le p < q \le |R|\};$ *if*  $r_p = r_q$ , then there exists  $< 0, 0 > \in \{< e_p^m, e_q^m >: 1 \le p < q \le |R|\}$  or (2)
- (3)

```
 \{ r_p \neq r_q, \text{ then there exists } < 1, 1 > \in \{ < e_p^m, e_q^m >: 1 \le p < q \le |R| \}; \\ if r_p \neq r_q, \text{ then there exists } < 1, 0 > \in \{ < e_p^m, e_q^m >: 1 \le p < q \le |R| \} \text{ and } \\ < 0, 1 > \in \{ < e_p^m, e_q^m >: 1 \le p < q \le |R| \}. 
(4)
```

# Proof.

- If there does not exist < 1,0 >  $\in \{ < e_p^m, e_q^m >: 1 \le p < q \le |R| \}$ , then < 0,0 >  $\in \{ < e_p^m, e_q^m >: 1 \le p < q \le |R| \}$  or < 0,1 >  $\in \{ < e_p^m, e_q^m >: 1 \le p < q \le |R| \}$  or (1) $<1,1>\in \{< e_p^m, e_q^m >: 1 \le p < q \le |R|\}$ . We have seven binary relations as follows:  $\{< 0, 0 >\}$ ,  $\{<0,1>\}, \{<1,1>\}, \{<0,0>,<0,1>\}, \{<0,0>,<1,1>\}, \{<0,1>,<1,1>\}$  and  $\{\langle 0,0\rangle,\langle 0,1\rangle,\langle 1,1\rangle\}$ . From seven binary relations above, if  $\forall a_m \in C, \exists a_n \in C,$ one cannot get  $e_p^m + e_q^m = e_p^m (e_p^n \neq e_q^n)$ . Thus, there exists  $< 1,0 > \in \{ < e_p^m, e_q^m >: 1 \le$  $p < q \leq |R|$  in  $M_{BDM}$ .
- If there does not exist  $< 0, 1 > \in \{< e_p^m, e_q^m >: 1 \le p < q \le |R|\}$ , then  $< 0, 0 > \in \{< e_p^m, e_q^m >: 1 \le p < q \le |R|\}$  or  $< 1, 0 > \in \{< e_p^m, e_q^m >: 1 \le p < q \le |R|\}$  or  $< 1, 1 > \in \{< e_p^m, e_q^m >: 1 \le p < q \le |R|\}$  or  $< 1, 1 > \in \{< e_p^m, e_q^m >: 1 \le p < q \le |R|\}$  or  $< 1, 1 > \in \{< e_p^m, e_q^m >: 1 \le p < q \le |R|\}$  or  $< 1, 1 > \in \{< e_p^m, e_q^m >: 1 \le p < q \le |R|\}$  or  $< 1, 1 < q < |R|\}$  or < 1, 1 < q < |R| or  $< 1, 1 < q < |R|\}$  or < 1, 1 < q < |R| or < 1, 1 < q < |R|(2)

 $1 \le p < q \le |R|$ . Thus, we can also have seven binary relations as follows:  $\{< 0, 0 >\}$ ,  $\{< 1, 0 >\}$ ,  $\{< 1, 1 >\}$ ,  $\{< 0, 0 >, < 1, 0 >\}$ ,  $\{< 0, 0 >, < 1, 1 >\}$ ,  $\{< 1, 0 >, < 1, 1 >\}$  and  $\{< 0, 0 >, < 1, 0 >, < 1, 1 >\}$ . From seven binary relations above, if  $\forall a_m \in C$  and  $\exists a_n \in C$ , one cannot get  $e_p^m + e_q^m = e_q^m (e_q^n \neq e_p^n)$ . Thus, there exists  $< 0, 1 > \in \{< e_p^m, e_q^m >: 1 \le p < q \le |R|\}$  in  $M_{BDM}$ .

- (3) If there does not exist  $< 0, 0 > \in \{< e_p^m, e_q^m >: 1 \le p < q \le |R|\}$  or  $< 1, 1 > \in \{< e_p^m, e_q^m >: 1 \le p < q \le |R|\}$  or  $< 1, 1 > \in \{< e_p^m, e_q^m >: 1 \le p < q \le |R|\}$ . There must have seven binary relations as follows:  $\{< 0, 1 >\}, \{< 1, 0 >\}, \{< 0, 1 >, < 1, 0 >\}, \{< 0, 0 >, < 0, 1 >\}, \{< 0, 0 >, < 1, 0 >\}, \{< 1, 1 >, < 0, 1 >\}, \{< 1, 1 >, < 0, 1 >\}, \{< 1, 1 >, < 0, 1 >\}, \{< 1, 1 >, < 0, 1 >\}, \{< 1, 1 >, < 0, 1 >\}, \{< 1, 1 >, < 0, 1 >\}, \{< 1, 1 >, < 0, 1 >\}, \{< 1, 1 >, < 0, 1 >\}, \{< 1, 1 >, < 0, 1 >\}, \{< 1, 1 >, < 0, 1 >\}, \{< 1, 1 >, < 0, 1 >\}, \{< 1, 1 >, < 0, 1 >\}, \{< 1, 0 >\}$ . From seven binary relations above, we cannot have  $e_p^m = e_q^m$ . Thus, there exists  $< 0, 0 > \in \{< e_p^m, e_q^m >: 1 \le p < q \le |R|\}$  or  $< 1, 1 > \in \{< e_p^m, e_q^m >: 1 \le p < q \le |R|\}$  in  $M_{BDM}$ .
- (4) If there does not exist  $< 1,0 > \in \{< e_p^m, e_q^m >: 1 \le p < q \le |R|\}$  and  $< 0,1 > \in \{< e_p^m, e_q^m >: 1 \le p < q \le |R|\}$ . We may obtain eleven binary relations as follows:  $\{< 0,0 >\}, \{<1,1>\}, \{<0,1>\}, \{<1,0>\}, \{<0,0>,<0,1>\}, \{<0,0>,<1,0>\}, \{<1,1>,<1,0>\}, \{<0,0>,<1,1>\}, \{<0,0>,<1,1>, \{<0,0>,<1,1>\}, \{<0,0>,<1,1>\}, \{<0,0>,<1,1>\}, \{<0,0>,<1,1>\}, \{<0,0>,<1,1>,<0,1>\}, \{<0,0>,<1,1>,<1,0>\}$ . From eleven binary relations, for  $\forall a_m \in C$ , we cannot have  $e_p^m \neq e_q^m$ . Thus, there exists  $< 1,0 > \in \{< e_p^m, e_q^m >: 1 \le p < q \le |R|\}$  and  $< 0,1 > \in \{< e_p^m, e_q^m >: 1 \le p < q \le |R|\}$  and  $< 0,1 > \in \{< e_p^m, e_q^m >: 1 \le p < q \le |R|\}$  in  $M_{BDM}$ .

This completes the proof.  $\Box$ 

Analogous to Theorem 1, we can easily obtain the following theorem as:

**Theorem 2.** Let  $M_{BDM} = (m_{BDM}(x_i, x_j))$  be a binary discernibility matrix, for  $\forall a_m, a_n \in C, \forall r_p \in R$ , we can have:

 $\begin{array}{ll} (1) & if \ a_m \supset a_n, \ then \ there \ exists < 1, 0 > \in \{ < e_p^m, e_p^n >: 1 \le m < n \le |C| \}; \\ (2) & if \ a_n \supset a_m, \ then \ there \ exists < 0, 1 > \in \{ < e_p^m, e_p^n >: 1 \le m < n \le |C| \}; \\ (3) & if \ a_m = a_n, \ then \ there \ exists < 0, 0 > \in \{ < e_p^m, e_p^n >: 1 \le m < n \le |C| \}; \\ (4) & if \ a_m \neq a_n, \ then \ there \ exists < 1, 0 > \in \{ < e_p^m, e_p^n >: 1 \le m < n \le |C| \}; \\ (4) & if \ a_m \neq a_n, \ then \ there \ exists < 1, 0 > \in \{ < e_p^m, e_p^n >: 1 \le m < n \le |C| \}; \\ (4) & a_m \neq a_n, \ then \ there \ exists < 1, 0 > \in \{ < e_p^m, e_p^n >: 1 \le m < n \le |C| \}; \\ (5) & a_m \neq a_n, \ then \ there \ exists < 1, 0 > \in \{ < e_p^m, e_p^n >: 1 \le m < n \le |C| \}. \\ \end{array}$ 

# Proof.

- (1) If there does not exist < 1,0 >  $\in$  {<  $e_p^m, e_p^n$  >: 1 ≤ m < n ≤ |C|}, then < 0,0 >  $\in$  {<  $e_p^m, e_p^n$  >: 1 ≤ m < n ≤ |C|} or < 0,1 >  $\in$  {<  $e_p^m, e_p^n$  >: 1 ≤ m < n ≤ |C|} or < 1,1 >  $\in$  {<  $e_p^m, e_p^n$  >: 1 ≤ m < n ≤ |C|} or < 1,1 >  $\in$  {<  $e_p^m, e_p^n$  >: 1 ≤ m < n ≤ |C|} or < 1,1 >  $\in$  {<  $e_p^m, e_p^n$  >: 1 ≤ m < n ≤ |C|}. We have seven binary relations as follows: {< 0,0 >}, {< 0,0 >}, {< 0,1 >}, {< 0,1 >}, {< 0,1 >}, {< 0,0 >, < 0,1 >}, {< 0,0 >, < 0,1 >}, {< 0,0 >, < 1,1 >}, {< 0,1 >, < 1,1 >} and {< 0,0 >, < 0,1 >, < 1,1 >}. From seven binary relations above, if  $\forall r_p \in R, \exists r_q \in R$ , one cannot get  $e_p^m + e_p^n = e_p^m(e_q^m \neq e_q^n)$ . Thus, there exists < 1,0 >  $\in$  {<  $e_p^m, e_p^n$  >: 1 ≤ m < n ≤ |C|} in  $M_{BDM}$ .
- (2) If there does not exist  $< 0, 1 > \in \{< e_p^m, e_p^n >: 1 \le m < n \le |C|\}$ , then  $< 0, 0 > \in \{< e_p^m, e_p^n >: 1 \le m < n \le |C|\}$  or  $< 1, 0 > \in \{< e_p^m, e_p^n >: 1 \le m < n \le |C|\}$  or  $< 1, 1 > \in \{< e_p^m, e_p^n >: 1 \le m < n \le |C|\}$  or  $< 1, 1 > \in \{< e_p^m, e_p^n >: 1 \le m < n \le |C|\}$  or  $< 1, 1 > \in \{< e_p^m, e_p^n >: 1 \le m < n \le |C|\}$  or  $< 1, 1 > \in \{< e_p^m, e_p^n >: 1 \le m < n \le |C|\}$ . Thus, we can also have seven binary relations as follows:  $\{< 0, 0 >\}$ ,  $\{< 1, 0 >\}$ ,  $\{< 1, 1 >\}$ ,  $\{< 0, 0 >, < 1, 0 >\}$ ,  $\{< 0, 0 >, < 1, 1 >\}$ ,  $\{< 1, 0 >, < 1, 1 >\}$  and  $\{< 0, 0 >, < 1, 0 >, < 1, 1 >\}$ . From seven binary relations above, if  $\forall r_p \in R$  and  $\exists r_q \in R$ , one cannot get  $e_p^m + e_p^n = e_p^n (e_q^m \neq e_q^n)$ . Thus, there exists  $< 0, 1 > \in \{< e_p^m, e_p^n >: 1 \le m < n \le |C|\}$  in  $M_{BDM}$ .
- (3) If there does not exist  $< 0, 0 > \in \{< e_p^m, e_p^n >: 1 \le m < n \le |C|\}$  or  $< 1, 1 > \in \{< e_p^m, e_p^n >: 1 \le m < n \le |C|\}$ . There must have seven binary relations as follows:  $\{< 0, 1 >\}, \{< 1, 0 >\}, \{< 0, 0 >, < 0, 1 >\}, \{< 0, 0 >, < 1, 0 >\}, \{< 1, 1 >, < 0, 1 >\}, \{< 1, 1 >, < 0, 1 >\}, \{< 1, 1 >, < 0, 1 >\}, \{< 1, 1 >, < 1, 0 >\}$ . From seven binary relations above, we cannot have  $e_p^m = e_p^n$ . Thus,

there must exists  $< 0,0 > \in \{< e_p^m, e_p^n >: 1 \le m < n \le |C|\}$  or  $< 1,1 > \in \{< e_p^m, e_p^n >: 1 \le m < n \le |C|\}$  in  $M_{BDM}$ .

(4) If there does not exist < 1,0 >  $\in$  {<  $e_p^m, e_p^n$  >: 1 ≤  $m < n \le |C|$ } and < 0,1 >  $\in$  {<  $e_p^m, e_p^n$  >: 1 ≤  $m < n \le |C|$ }. We may obtain eleven binary relations as follows: {< 0,0 >}, {< 1,1 >}, {< 0,1 >}, {< 1,0 >}, {< 0,0 >,< 0,1 >}, {< 0,0 >,< 1,0 >}, {< 1,1 >,< 0,1 >}, {< 1,0 >}, {< 0,0 >,< 1,1 >}, {< 0,0 >,< 1,1 >}, {< 0,0 >}, {< 1,1 >,< 0,1 >}, {< 0,0 >}, {< 1,1 >,< 0,1 >}, {< 0,0 >}, {< 1,1 >,< 0,1 >}, {< 0,0 >}, {< 1,1 >}, {< 0,0 >}, {< 1,1 >}, {< 0,0 >}, {< 1,1 >}, {< 0,0 >}, {< 1,1 >}, {< 0,1 >}, {< 0,0 >}, {< 1,1 >}, {< 0,1 >}, {< 0,0 >}, {< 1,1 >}, {< 0,1 >}, {< 0,0 >}, {< 1,1 >}, {< 0,1 >}, {< 0,0 >}, {< 1,1 >}, {< 0,1 >}, {< 0,0 >}, {< 1,1 >}, {< 0,1 >}, {< 0,0 >}, {< 1,1 >}, {< 0,1 >}, {< 0,0 >}, {< 1,1 >}, {< 0,1 >}, {< 0,0 >}, {< 1,1 >}, {< 0,1 >}, {< 0,0 >}, {< 1,1 >}, {< 0,1 >}, {< 0,0 >}, {< 1,1 >}, {< 0,1 >}, {< 0,0 >}, {< 1,1 >}, {< 0,1 >}, {< 0,0 >}, {< 1,1 >}, {< 0,1 >}, {< 0,0 >}, {< 1,1 >}, {< 0,1 >}, {< 0,0 >}, {< 1,1 >}, {< 0,1 >}, {< 0,0 >}, {< 0,0 >}, {< 1,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {< 0,0 >}, {

This completes the proof.  $\Box$ 

Deterministic finite automaton, also called deterministic finite acceptor, is an important concept in theory of computation. A deterministic finite automaton constructs a finite-state machine which can accept or reject symbol strings, and produce a computation of automation for each input string. In what follows, we adopt deterministic finite automata to obtain row relations and column relations in a binary discernibility matrix. Here, we first review the definition of deterministic finite automaton as follows.

**Definition 9.** A deterministic finite automaton is a 5-tuple  $(Q, \sum, \delta, S_0, F)$ , where Q is a finite nonempty set of states,  $\sum$  is a finite set of input symbols,  $\delta$  is a transition function,  $S_0 \in Q$  is a start state, F is a set of accept states.

Regarding object pair  $e_p^m e_q^m$  as the basic granule in input symbols, a deterministic finite automaton for row relations in a binary discernibility matrix is illustrated by the following theorem:

**Theorem 3.** A deterministic finite automaton for row relations, denoted by  $DFA_{row}$ , is a 5-tuple  $(Q, \sum, \delta, S_0, F)$ , where  $Q = \{S_0, S_1, S_2, S_3, S_4\}$  is a finite set of states,  $\sum = \{e_p^0 e_q^0 e_p^1 e_q^1 \dots e_p^m e_q^m e_p^{m+1} e_q^{m+1} \dots e_p^{|C|} e_q^{|C|}\}(1 < m < |C|, r_p, r_q \in R)$  is an input binary character string,  $\delta$  is a transition function,  $S_0 \in Q$  is a start state,  $F = \{S_1, S_2, S_3, S_4\}$  is a set of accept states. A deterministic finite automaton for row relations can be illustrated in Figure 1 as follows.



Figure 1. A Deterministic Finite Automaton for Row Relations.

**Proof.** In a binary discernibility matrix, relations between  $r_p$  and  $r_q$  can be concluded as  $r_q \supset r_p$ ,  $r_p = r_q$ ,  $r_p \supset r_q$  and  $r_p \neq r_q$ .

We discuss a deterministic finite automaton for row relations from four parts separately, as follows.

(1) According to Definition 5 and Theorem 1, for  $\forall a_m \in C, \forall r_p, r_q \in R$ , there must be  $< 0, 1 > \in \{< e_p^m, e_q^m >: 1 \le p < q \le |R|\}$ . Thus, the regular expression for  $r_q \supset r_p$  can be defined as

 $[(00/11)^*(01)^+(00/11)^*]^+$ . We can easily have the corresponding deterministic finite automaton in Figure 2 as:



**Figure 2.** A Deterministic Finite Automaton for  $r_q \supset r_p$ .

(2) For  $r_p = r_q$ , there must be  $< 0, 0 > \in \{ < e_p^m, e_q^m >: 1 \le p < q \le |R| \}$  or  $< 1, 1 > \in \{ < e_p^m, e_q^m >: 1 \le p < q \le |R| \}$ . The regular expression for  $r_p = r_q$  is denoted by  $(00/11)^+$ . So, the corresponding deterministic finite automaton can be illustrated in Figure 3 as:



**Figure 3.** A Deterministic Finite Automaton for  $r_p = r_q$ .

(3) Analogous to  $r_p \supset r_q$ , for  $r_p, r_q \in R$ , there must be  $< 1, 0 > \in \{< e_p^m, e_q^m >: 1 \le p < q \le |R|\}$ . Therefore, the regular expression for  $r_p \supset r_q$  can be obtained as  $[(00/11)^*(10)^+(00/11)^*]^+$ . We can easily have the corresponding deterministic finite automaton in Figure 4 as:



**Figure 4.** A Deterministic Finite Automaton for  $r_p \supset r_q$ .

(4) For  $r_p \neq r_q$ , there must be  $< 1, 0 > \in \{ < e_p^m, e_q^m >: 1 \le p < q \le |R| \}$  and  $< 0, 1 > \in \{ < e_p^m, e_q^m >: 1 \le p < q \le |R| \}$ . The regular expression for  $r_p \neq r_q$  is denoted by  $[(00/11)^*(01)^+(00/11)^*(10)^+(00/11)^*]^+/[(00/11)^*(10)^+(00/11)^*(01)^+(00/11)^*]^+$ . Hence, the corresponding deterministic finite automaton can be illustrated in Figure 5 as:



**Figure 5.** A Deterministic Finite Automaton for  $r_p \neq r_q$ .

One can construct a deterministic finite automaton for row relations by four deterministic finite automata shown in Figure 1.

This completes the proof.  $\Box$ 

Similar to the deterministic finite automaton for row relations, we present the deterministic finite automaton for column relations in a binary discernibility matrix as follows.

**Theorem 4.** A deterministic finite automaton for column relations  $DFA_{col}$  is a 5-tuple  $(Q, \sum, \delta, S_0, F)$ , where  $Q = \{S_0, S_1, S_2, S_3, S_4\}$  is a finite set of states,  $\sum = \{e_0^m e_0^n e_1^m e_1^n \dots e_p^m e_p^n e_{p+1}^n \dots e_{|R|}^m e_{|R|}^n\}$   $(1 is an input binary character string, <math>\delta$  is a transition function,  $S_0 \in Q$  is a start state.  $F = \{S_1, S_2, S_3, S_4\}$  is a set of accept states. A deterministic finite automaton for column relations can be illustrated in Figure 6 as follows:



Figure 6. A Deterministic Finite Automaton for Column Relations.

**Proof.** This proof is similar to the proof of Theorem 3.  $\Box$ 

By means of the proposed deterministic finite automata for row and column relations, we propose a quick algorithm for binary discernibility matrix simplification using deterministic finite automata (BDMSDFA) as follows:

We present the following example to explain Algorithm 3 as follows.

# Algorithm 3 : A quick algorithm for binary discernibility matrix simplification using deterministic finite automata, BDMSDFA.

**Input:** Original binary discernibility matrix  $M_{BDM}$ ; **Output:** Simplified binary discernibility matrix  $M'_{BDM}$ 

```
1: delete the row in which all elements are 0 s;
2: compare the row relation between r_p and r_q by DFA_{row}
   for p = 1 to |R| do
3:
       for q = 1 to |R| do
 4:
5:
           if r_p \supset r_q then
               delete row r_p from M_{BDM}
6:
              break
7.
           end if
8:
       end for
9:
10: end for
11: delete the column in which all elements are 0 s;
12: compare the column relation between a_m and a_n by DFA_{col}
13: for m = 1 to |C| do
       for n = 1 to |C| do
14:
15:
           if a_n \supset a_m then
              delete column a_m from M_{BDM}
16:
              break.
17:
           end if
18:
       end for
19:
20: end for
21: output a simplified binary discernibility matrix M'_{BDM};
```

**Example 1.** Let  $DS = (U, C \cup D, V, f)$  be a decision system shown Table 1, where the universe  $U = \{x_1, x_2, x_3, x_4, x_5\}$ , the condition attribute set  $C = \{a_1, a_2, a_3, a_4\}$ , the decision attribute set  $D = \{d\}$ .

	<i>a</i> <sub>1</sub>	<i>a</i> <sub>2</sub>	<i>a</i> <sub>3</sub>	$a_4$	d	
<i>x</i> <sub>1</sub>	1	2	0	0	1	
<i>x</i> <sub>2</sub>	1	2	0	0	1	
<i>x</i> <sub>3</sub>	1	2	1	1	2	
$x_4$	1	3	1	1	3	
$x_5$	2	4	1	1	4	

Table 1. A decision system.

For the decision system above, we have the corresponding binary discernibility matrix as follows:

$$M_{BDM}^{1} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

We delete the row in which all elements are 0 s in  $M_{BDM}^1$ , and obtain the binary discernibility matrix  $M_{BDM}^2$  as follows.

$$M_{BDM}^2 = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}.$$

In the binary discernibility matrix  $M_{BDM}^2$ ,  $r_1 : 0011$ ,  $r_2 : 0111$ ,  $r_3 : 1111$ ,  $r_4 : 0011$ ,  $r_5 : 0111$ ,  $r_6 : 1111$ ,  $r_7 : 0100$ ,  $r_8 : 1100$ ,  $r_9 : 1100$ . According to the definition of the deterministic finite automaton for row relations, we have  $\sum_{12} = 00011111$ ,  $\sum_{13} = 01011111$ ,  $\sum_{14} = 00001111$ ,  $\sum_{15} = 00011111$ ,  $\sum_{16} = 01011111$ ,  $\sum_{17} = 00011010$ ,  $\sum_{18} = 01011010$ ,  $\sum_{19} = 01011010$ . By using the deterministic finite automaton for row relations shown in Figure 1, we can get the row relations as follows.  $r_2 \supset r_1$ ,  $r_3 \supset r_1$ ,  $r_4 = r_1$ ,  $r_5 \supset r_1$ ,  $r_6 \supset r_1$ ,  $r_7 \neq r_1$ ,  $r_8 \neq r_1$ ,  $r_9 \neq r_1$ . Therefore, we delete  $r_2$ ,  $r_3$ ,  $r_5$  and  $r_6$ . Similarly, we get  $r_8 \supset r_7$  and  $r_9 \supset r_7$ ,  $r_4 \neq r_7$ , and then delete  $r_8$  and  $r_9$ . Therefore, we have the following binary discernibility matrix:

$$M_{BDM}^3 = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix}.$$

We delete the column in which all elements are  $0 \text{ s in } M_{BDM}^3$ , and have

$$M_{BDM}^4 = egin{pmatrix} 0 & 1 & 1 \ 0 & 1 & 1 \ 1 & 0 & 0 \end{pmatrix}.$$

In the binary discernibility matrix  $M_{BDM}^4$ ,  $a_1 : 001$ ,  $a_2 : 110$ ,  $a_3 : 110$ . According to the definition of deterministic finite automaton for column relations, we have  $\sum_{12} = 010110$ ,  $\sum_{13} = 010110$ ,  $\sum_{23} = 111100$ . By using the deterministic finite automaton for column relations shown in Figure 6, we have  $a_1 \neq a_2$ ,  $a_1 \neq a_3$ ,  $a_2 = a_3$ . Thus, we cannot delete any column in  $M_{BDM}^4$ , and get the following binary discernibility matrix.

$$M_{BDM}^5 = \begin{pmatrix} 0 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

A  $10 \times 4$  matrix  $M_{BDM}^1$  is compressed to a  $3 \times 3$  matrix  $M_{BDM}^5$ . The simplified binary discernibility matrix with fewer objects or columns will be help in improving the efficiency of attribute reduction.

Assume that t = |R| and s = |C|, the upper bound of time complexity of BDMR is 3ts(t + s - 2), the lower bound of time complexity of BDMR is 2ts(t + s - 2). The upper bound of time complexity of IBDMR is 3ts(t + s - 2) and the lower bound of the worst-case time complexity of IBDMR is 2ts(t + s - 2). By employing deterministic finite automata, the algorithm complexity of BDMSDFA is ts(t + s - 2). Obviously, the time complexity of BDMSDFA is lower than that of BDMR and IBDMR. Therefore, it is concluded that the proposed algorithm BDMSDFA reduces the computational time for binary discernibility matrix simplification in general.

The advantages of the proposed method are expressed as follows. (1) Deterministic finite automata in a binary discernibility matrix are constructed, it can provide an understandable approach to comparing the relationships of different rows (columns) quickly. (2) Based on deterministic finite automata, a high efficiency algorithm of binary discernibility matrix simplification is developed. Theoretical analyses and experimental results indicate that the proposed algorithm is effective and efficient. It should be noted that the proposed method is based on Pawlak decision systems, but not suitable for generalized decision systems, such as incomplete decision systems, interval-valued decision systems and fuzzy decision systems. Deterministic finite automata in generalized decision systems will be investigated in the future.

#### 5. Experimental Results and Analyses

The objective of the following experiments in this section is to demonstrate the high efficiency of the algorithm BDMSDFA. The experiments are divided into two aspects. In one aspect, we employ 10 datasets in Table 2 to verify the performance of time consumption of BDMR, IBDMR and BDMSDFA. In the other aspect, the computational times of algorithms BDMR, IBDMR and BDMSDFA with the increase of the size of attributes (or objects) are calculated respectively. We carry out three algorithms on a personal computer with Windows 8.1 (64 bit) and Inter(R) Core(TM) i5-4200U, 1.6 GHz and 4 GB memory. The software is Microsoft Visual Studio 2017 version 15.9 and C++. Data sets used in the experiments are all downloaded from UCI repository of machine learning data sets (http://archive.ics.uci.edu/ml/datasets.html).

Table 2 indicates the computational time of BDMR, IBDMR and BDMSDFA on the 10 data sets. We can see that the algorithm BDMSDFA is much faster than the algorithms BDMR and IBDMR. The computational times of three algorithms follows this order:  $BDMR \ge IBDMR > BDMSDFA$ . The computational time of BDMSDFA is the minimum among the three algorithms. For the data set Auto in Table 2, the computational times of BDMR and IBDMR and IBDMR are 75 ms and 68 ms, while that of BDMSDFA is 36 ms. For the data set Credit\_a, the computational times of BDMR and IBDMR are

113 ms and 105 ms, while that of BDMSDFA is 55 ms. For some data sets in Table 2, the computational time of BDMSDFA can reduce over half the computational time of BDMR or IBDMR. In Table 2, for the data set Breast\_w, the computational times of BDMR and IBDMR are 75 ms and 73 ms, while that of BDMSDFA is 29 ms. For the data set Promoters, the computational times of BDMR and IBDMR are 1517 ms and 936 ms, while that of BDMSDFA is only 398 ms. For the date sets such as Lung-cancer, Credit\_a, Breast\_w, Anneal, the computational time of BDMR is close to that of IBDMR. For the data set I abor\_neg, the computational time of BDMR is relatively smaller than difference between BDMR (IBDMR) and BDMSDFA.

Data Sets	Num. of Objects	Num. of Attributes	Num. of Rows	Num. of Columns	Time of BDMR (ms)	Time of IBDMR (ms)	Time of BDMSDFA (ms)
Labor_neg	40	15	52	14	2	2	1
Lung-cancer	32	57	206	53	11	9	3
Heart_statlog	270	14	96	13	32	27	14
Autos	250	26	36	19	75	68	36
Credit_a	690	16	27	12	113	105	55
Breast_w	699	10	20	9	75	73	29
Anneal	898	39	9	9	206	194	134
Promoters	106	58	2761	57	1517	936	398
Dermatology	366	35	1347	31	3239	2638	1318
Connect_4	67,557	43	697	42	2,444,328	2,433,863	1,759,917

Table 2. Time consumption of BDMR, IBDMR and BDMSDFA.

We compare the computational times of BDMR, IBDMR and BDMSDFA with the increase of the size of objects. In Figure 7a–f, the x-coordinate pertains to the size of objects in the universe, while the y-coordinate concerns the time consumption of algorithms. We employ 6 data sets (Dermatlogy, Credit\_a, Controceptive\_Method\_Choice, Letter, Flag and Mushroom) to verify the performance of time consumption of BDMR, IBDMR and BDMSDFA. When dealing with the same UCI data sets, the computational time of BDMSDFA is less than that of BDMR and IBDMR, in other words, BDMSDFA is more efficient than BDMR and IBDMR. Figure 7 shows more detailed change trends of each algorithm with the number of objects increasing. The computational times of three algorithms increase with the increase of the number of objects simultaneously. It is obvious to see that the slope of the curve of BDMSDFA is smaller than the curve of BDMR or IBDMR, and the computational time of BDMSDFA increases slowly. The differences between BDMR (IBDMR) and BDMSDFA become distinctly larger when the size of the objects increases. In Figure 7c, the difference of BDMR (IBDMR) and BDMSDFA is not obviously different at the beginning. The computational time of DBMR (IBDMR) increases distinctly when the number of objects is over 450. The computational time of algorithm BDMR increases by 479 ms when the number of objects rises from 450 to 1473, whereas the computational time of algorithm BDMSDFA increases by only 141 ms. In Figure 7e, the computational time of the algorithm IBDMR increases by 104 ms when the number of objects rises from 20 to 160, whereas the time consumption of algorithm BDMSDFA increases by only 49.





Figure 7. (a) Dermatlogy; (b) Credit\_a; (c) Controceptive\_Method\_Choice; (d) Letter; (e) Flag; (f) Mushroom.

In Figure 8a–f, the *x*-coordinate pertains to the size of attributes, while the *y*-coordinate concerns the time consumption of algorithms. We also take 6 data sets (Dermatlogy, Credit\_a, Controceptive\_Method\_Choice, Letter, Flag and Mushroom) to verify the performance of the computational times of BDMR, IBDMR and BDMSDFA. The curve of BDMR is similar to that of IBDMR. The curve of BDMSDFA is under the curves of BDMR and IBDMR. Then, the computational time of BDMSDFA is less than that of BDMR or IBDMR. In Figure 8b, the computational time of algorithms BDMR and IBDMR increase by 164 ms and 123 ms respectively, while the computational

time of algorithm BDMSDFA increases by 58 ms. In Figure 8c, the curves of BDMR and IBDMR raise profoundly when the size of the attributes increases. In Figure 8e, the computational time of algorithm IBDMR increases from 4 ms to 105 ms when the number of objects rises from 3 to 24, while the computational time of algorithm BDMSDFA increasedly from 2 ms to 50 ms. For Figure 8a–f, it is concluded that the efficiency of BDMSDFA is higher than that of BDMR or IBDMR with the increase of the number of attributes. Difference between BDMR and IBDMR is relatively smaller than difference between BDMR (IBDMR) and BDMSDFA. The computational times of three algorithms increase with the increase of the number of attributes monotonously. When dealing with the same situation, the computational time of BDMSDFA is the minimum among the three algorithms.



Figure 8. (a) Dermatlogy; (b) Credit\_a; (c) Controceptive\_Method\_Choice; (d) Letter; (e) Flag; (f) Mushroom.

Experimental analyses and results show a high efficiency of the algorithm BDMSDFA. The proposed simplification algorithm using deterministic finite automata can be applied as a preprocessing technique for data compression and attribute reduction in large-scale data sets.

# 6. Conclusions

Original binary discernibility matrices which are not simplified usually have irrelative objects and attributes. These irrelative objects and attributes may lead to inefficiency in attribute reduction, knowledge acquisition, etc. To tackle this problem, a quick method of comparing the relationships of different rows (columns) are introduced in binary discernibility matrices. By using deterministic finite automata, a quick algorithm for binary discernibility matrix simplification (BDMSDFA) is developed. The experiment results indicate that DBMSDFA can get higher performance in the efficiency of binary discernibility matrix simplification. The contributions of this paper can be summarized as follows.

- (1) We define row (or column) relations which are used for constructing deterministic finite automata.
- (2) Deterministic finite automata are firstly used for comparing the relationships of different rows (columns) in a binary discernibility matrix.
- (3) Based on deterministic finite automata, a quick algorithm for binary discernibility matrix simplification is developed. Experimental results indicate that the relationship between the time consumption of BDMSDFA and the number of objects (attributes) is strictly monotonic. With the increase of the size of objects (attributes), the algorithm BDMSDFA is more efficient than BDMR and IBDMR.

It is noted that the proposed quick simplification algorithm for discernibility matrix is only suitable for completed decision systems. However, in practical applications, there exists many generalized decision systems, such as incomplete decision systems, interval-valued decision systems, etc. Researches on quick simplification algorithms in generalized decision systems will be investigated. Combing the researches on fuzzy sets [33–36], we will propose the fuzzy binary discernibility matrix. Some applications of the (fuzzy) binary discernibility matrix simplification will also be studied in the future.

**Author Contributions:** N.Z. and B.L. designed the research. N.Z gave theoretical analysis and wrote the whole paper. N.Z. and B.L. designed and conducted the experiments. Z.Z. and Y.G. provided some suggestions on the paper.

**Funding:** This research was funded by the National Natural Science Foundation of China (No. 61403329, No. 61572418, No. 61502410, No. 61572419).

Acknowledgments: The authors thank the anonymous referees for the constructive comments and suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

# References

- 1. Arora, R.; Garg, H. A robust correlation coefficient measure of dual hesitant fuzzy soft sets and their application in decision making. *Eng. Appl. Artif. Intell.* **2018**, *72*, 80–92. [CrossRef]
- 2. Garg, H.; Nancy. New Logarithmic operational laws and their applications to multiattribute decision making for single-valued neutrosophic numbers. *Cogn. Syst. Res.* **2018**, *52*, 931–946. [CrossRef]
- 3. Garg, H.; Nancy. Linguistic single-valued neutrosophic prioritized aggregation operators and their applications to multiple-attribute group decision-making. *J. Ambient Intell. Hum. Comput.* **2018**, *9*, 1975–1997. [CrossRef]
- 4. Garg, H.; Arora, R. Dual hesitant fuzzy soft aggregation operators and their application in decision making. *Cogn. Comput.* **2018**, *10*, 769–789. [CrossRef]
- Garg, H.; Kumar, K. An advanced study on the similarity measures of intuitionistic fuzzy sets based on the set pair analysis theory and their application in decision making. *Soft Comput.* 2018, 22, 4959–4970. [CrossRef]

- 6. Garg, H.; Kumar, K. Distance measures for connection number sets based on set pair analysis and its applications to decision-making process. *Appl. Intell.* **2018**, *48*, 3346–3359. [CrossRef]
- 7. Badi, I.; Ballem, M. Supplier selection using the rough BWM-MAIRCA model: a case study in pharmaceutical supplying in Libya. *Decis. Mak. Appl. Manag. Eng.* **2018**, *1*, 16–33. [CrossRef]
- Chatterjee, K.; Pamucar, D.; Zavadskas, E.K. Evaluating the performance of suppliers based on using the R'AMATEL-MAIRCA method for green supply chain implementation in electronics industry. *J. Clean. Prod.* 2018, 184, 101–129. [CrossRef]
- 9. Pawlak, Z. Rough sets. Int. J. Inf. Comput. Sci. 1982, 11, 341–356. [CrossRef]
- Felix, R.; Ushio, T. Rough sets-based machine learning using a binary discernibility matrix. In Proceedings of the Second International Conference on Intelligent Processing and Manufacturing of Materials, Honolulu, HI, USA, 10–15 July 1999; pp. 299–305.
- 11. Ding, M.W.; Zhang, T.F.; Ma, F.M. Incremental attribute reduction algorithm based on binary discernibility matrix. *Comput. Eng.* **2017**, *43*, 201–206.
- 12. Qian, W.B.; Xu, Z.Y.; Yang, B.R.; Huang, L.Y. Efficient incremental updating algorithm for computing core of decision table. *J. Chin. Comput. Syst.* **2010**, *31*, 739–743.
- 13. Wang, Y.Q.; Fan, N.B. Improved algorithms for attribute reduction based on simple binary discernibility matrix. *Comput. Sci.* **2015**, *42*, 210–215.
- Zhang, T.F.; Yang, X.X.; Ma, F.M. Algorithm for attribute relative reduction based on generalized binary discernibility matrix. In Proceedings of the 26th Chinese Control and Decision Conference, Changsha, China, 30 May–2 June 2014; pp. 2626–2631.
- 15. Li, J.; Wang, X.; Fan, X.W. Improved binary discernibility matrix attribute reduction algorithm in customer relationship management. *Procedia Eng.* **2010**, *7*, 473–476. [CrossRef]
- 16. Tiwari, K.S.; Kothari, A.G.; Keskar, A.G. Reduct generation from binary discernibility matrix: an hardware approach. *Int. J. Future Comput. Commun.* **2012**, *1*, 270–272. [CrossRef]
- 17. Zhi, T.Y.; Miao, D.Q. The binary discernibility matrix's transformation and high Efficiency attributes reduction algorithm's conformation. *Comput. Sci.* **2002**, *29*, 140–142.
- 18. Yang, C.J.; Ge, H.; Li, L.S. Attribute reduction of vertically partitioned binary discernibility matrix. *Control Decis.* **2013**, *28*, 563–568.
- 19. Ren, Q.; Luo, Y.T.; Yao, G.S. An new method for modifying binary discernibility matrix and computation of core. *J. Chin. Comput. Syst.* **2013**, *34*, 1437–1440.
- 20. Ding, M.W.; Zhang, T.F.; Ma, F.M. Incremental attribute reduction algorithm based on binary discernibility matrix in incomplete information System. *Comput. Sci.* **2017**, *44*, 244–250.
- 21. Hu, S.P.; Zhang, Q.H.; Yao, L.Y. Effective algorithm for computing attribute core based on binary representation. *Comput. Sci.* 2016, *43*, 79–83.
- 22. Skowron, A.; Rauszer, C. The discernibility matrices and functions in information systems. In *Intelligent Decision Support: Handbook of Applications and Advances of the Rough Sets Theory;* Słowiński, R., Ed.; Kluwer Academic Publishers: Dordrecht, The Netherlands, 1992; pp. 331–362.
- 23. Guan, Y.Y.; Wang, H.K.; Wang, Y.; Yang, F. Attribute reduction and optimal decision rules acquisition for continuous valued information systems. *Inf. Sci.* 2009, 179, 2974–2984. [CrossRef]
- 24. Sun, B.Z.; Ma, W.M.; Gong, Z.T. Dominance-based rough set theory over interval-valued information systems. *Expert Syst.* **2014**, *31*, 185–197. [CrossRef]
- 25. Qian, Y.H.; Liang, J.Y.; Dang, C.Y. Interval ordered information systems. *Comput. Math. Appl.* 2008, 56, 1994–2009. [CrossRef]
- 26. Qian, Y.H.; Dang, C.Y.; Liang, J.Y.; Tang, D.W. Set-valued ordered information systems. *Inf. Sci.* 2009, 179, 2809–2832. [CrossRef]
- 27. Zhang, H.Y.; Leung, Y.; Zhou, L. Variable-precision-dominance-based rough set approach to interval-valued information systems. *Inf. Sci.* **2013**, 244, 75–91. [CrossRef]
- 28. Kryszkiewicz, M. Rough set approach to incomplete information systems. *Inf. Sci.* **1998**, *112*, 39–49. [CrossRef]
- 29. Leung, Y.; Li, D.Y. Maximal consistent block technique for rule acquisition in incomplete information systems. *Inf. Sci.* **2003**, *153*, 85–106. [CrossRef]
- 30. Guan, Y.Y.; Wang, H.K. Set-valued information systems. Inf. Sci. 2006, 176, 2507–2525. [CrossRef]

- 31. Miao, D.Q.; Zhao, Y.; Yao, Y.Y.; Li, H.; Xu, F. Relative reducts in consistent and inconsistent decision tables of the Pawlak rough set model. *Inf. Sci.* **2009**, *179*, 4140–4150. [CrossRef]
- 32. Wang, X.; Ma, Y.; Wang, L. Research on space complexity of binary discernibility matrix. *J. Tianjin Univ. Sci. Technol.* **2006**, *21*, 50–53. [CrossRef]
- 33. Kaur, G.; Garg, H. Generalized cubic intuitionistic fuzzy aggregation operators using t-norm operations and their applications to group decision-making process. *Arab. J. Sci. Eng.* **2018**. [CrossRef]
- 34. Garg, H. New logarithmic operational laws and their aggregation operators for Pythagorean fuzzy set and their applications. *Int. J. Intell. Syst.* **2019**, *34*, 82–106. [CrossRef]
- 35. Garg, H.; Singh, S. A novel triangular interval type-2 intuitionistic fuzzy sets and their aggregation operators. *Iran. J. Fuzzy Syst.* **2018**, *15*, 69–93.
- 36. Garg, H.; Kumar, K. Improved possibility degree method for ranking intuitionistic fuzzy numbers and their application in multiattribute decision making. *Granul. Comput.* **2018**. [CrossRef]



 $\odot$  2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).