Journal of Computer Science and Cybernetics, V.41, N.1 (2025), 95-103
DOI: 10.15625/1813-9663/22140

ON IRREDUNDANT INCOMPLETE DECISION TABLE

DINH THU KHANH[1,5], VU DUC THI[2], JANOS DEMETROVICS[3], PHAM VIET ANH[1,4,*]

[1] *Graduate University of Science and Technology, Vietnam Academy of Science and Technology, 18 Hoang Quoc Viet Street, Cau Giay District, Ha Noi, Viet Nam*
[2] *Information Technology Institute, Vietnam National University, 144 Xuan Thuy Street, Cau Giay District, Ha Noi, Viet Nam*
[3] *Institute for Computer and Control, Hungarian Academy of Sciences, Budapest, Kende, Hungary*
[4] *HaUI Institute of Technology, Hanoi University of Industry, 298 Cau Dien Street, Bac Tu Liem District, Ha Noi, Viet Nam*
[5] *Faculty of Information Technology, Electric Power University, 235 Hoang Quoc Viet Street, Bac Tu Liem District, Ha Noi, Viet Nam*

Abstract. In rough set (RS) theory, the issues raised by attribute reduction are significant [1]. These issues currently garner widespread interest and extensive research from scientists worldwide. To investigate the problem of incomplete decision tables (IDTs), Kryszkiewicz developed the notion of relative relation (also known as tolerance relation) [2]. In this study, researchers address algorithmic issues related to attribute reduction in IDTs using a relational database approach. In [3], the authors proved that the result of the Sperner system (SS) is equivalent to the reduct set within the incomplete decision table. Specifically, the set of reducts in an IDT is a Sperner system, and conversely, for a given Sperner system $\mathcal{K}$, there exists an IDT such that the set of its reducts is exactly $\mathcal{K}$. In this paper, we propose the concept of an irredundant incomplete decision table based on objects from an IDT. This means that in an IDT, some rows are considered redundant and need to be removed. We present an efficient algorithm with a worst-case complexity that is polynomial in the number of columns and rows of the decision table. The algorithm allows us to find an irredundant incomplete decision table from a given IDT.

Keyword. The reduct, tolerance relation, rough set theory, irredundant incomplete decision table.

1. INTRODUCTION

A decision table (DT) is an case of an information system used to organize data. In a decision table (DT), attribute reduction refers to eliminating redundant attributes from a set of condition attributes without losing important information from the DT. In addition, attribute reduction also brings ease of understanding to the rules and greatly improves the

*Corresponding author.
E-mail addresses: khanhdt@epu.edu.vn (D. T. Khanh); vdthi@vnu.edu.vn (V. D. Thi); janos.demetrovics@sztaki.hu (J. Demetrovics); anhpv@haui.edu.vn, vietanh.hn.4078@gmail.com (P. V. Anh).

classification ability of machine learning models. Up to now, attribute reduction methods following the approach of rough set theory have attracted a lot of attention from scientists around the world. Nevertheless, these methods primarily rely on a heuristic approach to identify a reduct within the complete decision table, based on specific evaluation criteria, while maintaining polynomial computational complexity. Compared to solely searching for a reduct, algorithms that identify reducts within the decision table can achieve an optimal reduct and are highly efficient. Consequently, the investigation into these algorithms holds considerable importance.

Today, decision tables frequently lack information, leading to their categorization as IDTs. The challenge of identifying a reduct within an IDT is regarded as a general issue. Some algorithms either eliminate objects with missing data or substitute missing values with others to address the problem of IDTs in knowledge mining tasks. Nevertheless, these methods lead to the loss of critical information or the substitution of data that may impact the table's consistency. As a result, these methods often fail to achieve high efficiency. To address these challenges, Kryszkiewicz [2] presented the concept of a tolerance relation and developed a tolerance rough set model to investigate the issue of IDTs. According to this approach, the authors in [4] used Boolean reasoning methodology and proposed a method for determining abbreviated attributes in an IDT (attributes involved in at least one reduction). Based on the approach using the relational data model [5, 6], authors in [7, 8] proposed a method with polynomial time for extracting all the reducts in the consistent decision table. Through this strategy, we might decrease the number of columns (attributes) in the DT. However, these methods still have computational limitations due to having to deal with inconsistent objects. We suggested a polynomial technique in [9] that shortens the decision table's lines (or objects). Therefore, we may eliminate unnecessary columns and rows from the DT using these two effective techniques. On the other side, it's crucial to identify every gap in the whole DT. The authors of [10] developed a technique for locating all shortenings in a consistent and comprehensive DT. The authors showed that the difficulty of locating every shortcut on this DT increases exponentially with the number of attributes. It implies that the following must be demonstrated: There is an exponential algorithm that identifies all these reductions, demonstrating that this problem's complexity is at least exponential.

A polynomial algorithm can discover a reduction on a perfectly consistent DT, but it is impossible to find a reduction with the smallest force. Therefore, there isn't a polynomial algorithm that handles this task yet. We showed in [11] that the set of reducts on a complete consistent decision table is equivalent to the SS - a combinatorial system in which the components do not comprise each other. In other words, studying the set of reductions can lead to studying the Sperner system. Based on the relational database approach, this study proposes an algorithm aimed at reducing the rows of a table. Specifically, our algorithm will eliminate redundant objects and retain only those that play a significant role in improving the effectiveness of classification models. The paper consists of four main sections. Section 2 will introduce basic concepts related to incomplete decision tables and Sperner systems, along with their properties and interrelationships. Section 3 will present an object reduction algorithm for incomplete decision tables, which results in an incomplete yet inconsistent decision table. Several illustrative examples of the proposed algorithm will also be clarified. Finally, Section 4 will discuss the issues raised by this research.

2. PRELIMINARY

Some fundamental ideas in rough set theory will be discussed in this part [9, 12, 13, 14, 15, 16, 17]. First, the decision table has four components $\mathcal{KS} = (U, C \cup D, V, f)$, in which $U = \{u_1, u_2, \ldots u_n\}$ is a finite nonempty set of objects, $C = \{c_1, c_2, \ldots . c_m\}$ is the set of condition attributes, D is the decision attribute set, satisfying $C \cap D = \emptyset$. For each attribute $a \in A = C \cup D$, $V_{\{a\}}$ is the value set of the attribute a with $V = \bigcup\limits_{a \in A} V_{\{a\}}$ and $f : U \times (A) \to V$ is called the information function, where $f(u, a) \in V_{\{a\}}$, $\forall u \in U$. The assumption is that D only consists of one decision attribute d, which can be reduced to a single attribute by utilizing encryption if D contains multiple attributes [10]. In consideration of this, we evaluate the decision table $\mathcal{KS} = (U, C \cup \{d\}, V, f)$, where $d \notin C$.

Next, given a decision table $\mathcal{KS} = (U, C \cup D, V, f)$, each attribute subset $P \subseteq C \cup \{d\}$ determines an indiscernibility relation $IND(P) = \{(x, y) \in U \times U : \forall c \in P, f(x, c) = f(y, c)\}$. Then, $IND(P)$ creates a partition $U/P = \{P_1, P_2, \ldots, P_m\}$ on U, where an equivalence class is an element in U/P.

The set $\overline{R}X$ is the upper approximation of X, where $\overline{R}X = \{u \in U : [u]_R \cap X \neq \emptyset\}$, the set $R\underline{X}$ is the lower approximation of X, where $R\underline{X} = \{u \in U : [u]_R \subseteq X\}$. From this, $\overline{R}X/R\underline{X}$ is the boundary of X and the positive region of $\{d\}$ is the set $POS_R(d) = \bigcup_{X \in U/d}(R\underline{X})$ with $R \subseteq C$ and $X \subseteq U$. It is evident that $POS_C(d) = U$ or the functional dependency $C \to d$ is true, then $\mathcal{KS}$ is a consistent decision table. If $\mathcal{KS}$ is inconsistent, then $POS_C(d)$ is the largest subset of U such that the functional dependency $C \to d$ is true.

Definition 1. Let $\mathcal{KS} = (U, C \cup \{d\}, V, f)$ be a decision table. If $P \subseteq C$ satisfies $POS_P(d) = POS_C(d)$, and $\forall P' \subset P, POS_{P'}(d) \neq POS_C(d)$, then P is a reduct of C.

It is easy to see that if $\mathcal{KS}$ is a consistent decision table, Definition 1 demonstrates that if $P \to d$ and $\forall P' \subset P, P' \not\to d$, then P is a reduct of $\mathcal{KS}$, and P_d denotes the set of all reducts of $\mathcal{KS}$.

Definition 2. Let r be a relation on $A = \{a_1, ..., a_n\}$. Then, $\forall a_i \in A$, there exists $\mathcal{D}_{a_i}$ such that $* \in \mathcal{D}_{a_i}$, and $h_j : A \to \cup \mathcal{D}_{a_i}$ satisfies $h_j(a_i) \in \mathcal{D}_{a_i}$.

Definition 3. Let r be a relation on $A = \{a_1, ..., a_n\}, B \subseteq A$. If each $a \in A$ in this case is a member of B, we denote $h_i \sim h_j(B)$ as follows: $h_i(a) = h_j(a)$, or $h_i(a) = *$, or $h_j(a) = *$.

Definition 4. Let r be a relation on $A = \{a_1, ..., a_n\}$. Then, $E, G \subseteq A$ and E tolerance determines G, denoted by $E \xrightarrow{t} G$ if: $(\forall h_i, h_j \in r)\,(h_i \sim h_j(E)$ then $h_i \sim h_j(G))$.

Set $T_r = (E, G) : E, G \subseteq A$ and $E \xrightarrow{t} G$. It is possible to observe that

1) $(E, G) \in T_r \forall E \subseteq R$.
2) $(E, G) \in T_r$, then $E \subseteq C, D \subseteq G$ has $(C, D) \in T_r$.
3) $(E, G) \in T_r, (G, C) \in T_r \implies (E, C) \in T_r$.

Set $E^+ = a \in R : E \xrightarrow{t} a$.

Definition 5. Given an IDT $\mathcal{KS} = (U, C \cup \{d\}, V, f)$ with $* \notin \mathcal{D}_d$, when it contains the character $*$ in the decision attribute d, it is a consistent incomplete decision table, where C is a set of condition attributes.

For the method using the tolerance relation presented in [2], the author proposed that the value of the decision attribute column d is unique in the consistent incomplete decision table. However, for our study, Definition 4 does not require the decision value to be unique. Therefore, our study is more general than the method proposed in [2]. If the $\mathcal{KS}$ is an IDT, we can examine the components of U using a polynomial time approach to remove components that don't make the $\mathcal{KS}$ consistent. When anything is removed, we ontain the set $U^{'}$, and $\mathcal{KS} = (U^{'}, C \cup \{d\}, V, f)$ is consistent.

Definition 6. Given an inconsistent IDT $\mathcal{KS} = (U, C \cup \{d\}, V, f)$, B is a reduct of $\mathcal{KS}$ if: $B \subseteq C : B \xrightarrow{t} d$ and $\nexists B' \subset B$ then $B' \not\rightarrow d$ (meaning B' is a true subset of B, then B' is not relative determination of d).

Set $PREX(C) = \{B$ is the set of reducts of $\mathcal{KS}\}$.

Definition 7. Assume that $A = \{a_1, ..., a_n\}$ and $\mathcal{K} = \{K_1, ..., K_m\}$ is a Sperner system (SS) on R if: $K_i \nsubseteq K_j,\ \forall j, i$.

Definition 8. [18] Given a SS $\mathcal{K} = \{K_1, ..., K_m\}$ on R, we define $\mathcal{K}^{-1}$ is the set of antikeys of $\mathcal{K}$ as follows

Set $\mathcal{K}^{-1} = \{A \subseteq R : (B \in \mathcal{K}) \implies B \nsubseteq K$ and $B \subseteq C)$ then $\exists A \in \mathcal{K} : A \subseteq C\}$,

$$\mathcal{K}^{-1} = \{A \subseteq R : (B \in \mathcal{K} \implies B \nsubseteq A \text{ and } A \subseteq C) \text{ then } \exists B \in \mathcal{K} : B \subseteq C\}.$$

It is evident that $\mathcal{K}^{-1}$ as one of the subsets of R, excludes the elements of $\mathcal{K}$, essentially representing the largest non-key set. Notably, $\mathcal{K}^{-1}$ is also regarded as SS. If a minimum key set exists, an anti-key set will also exist.

Remark 1. *Let $\mathcal{KS} = (U, C \cup \{d\}, V, f)$ be a consistent IDT. Set $r = U = \{r_1, ..., r_n\}$, $R = C \cup \{d\}$. As a result, $PREX(C) = C$. $K_d^t = \{A \subseteq C : A \to d$ and $\forall A' :\xrightarrow{t} \{d\}$ and $A' \nsubseteq A\}$ and $PREX(C)$ is the SS.*

Theorem 1. [19] *Let $\mathcal{KS} = (U, C \cup \{d\}, V, f)$ be a consistent IDT. Set $r = U = \{r_1, ..., r_n\}$ and $R = C \cup \{d\}$. We calculate the set of equals from r using $\varepsilon_r = \{E_{ij}\}$ with*

$$E_{ij} = \{a \in R : a(u_i) = a(u_j) \text{ or } a(u_i) = * \text{ or } a(u_j) = *\},\ i = 1, ..., m,\ j = 1, ..., m,\ i \leq j.$$

From ε_R set $M_d = \{B \in \varepsilon_r : B \neq R, d \notin B$ and $\nexists B' \in \varepsilon_r : d \notin B'$ and $B \subset B'\}$, then we have $M_d = (K_d^t)^{-1}$.

3. ALGORITHM TO FINDING IRREDUNDANT INCOMPLETE DECISION TABLE

In this section, we present an approach to generate a consistent incomplete decision table with a non-redundant attribute set.

Remark 2. *Let $\mathcal{KS} = (U, C \cup \{d\}, V, f)$ be a consistent IDT. Then, the calculation of M_d is a polynomial in terms of the size of U and C.*

For the decision table $\mathcal{KS} = (U, C \cup \{d\}, V, f)$ is a consistent IDT, we call $\mathcal{KS}$ is an irredundant consistent decision table if $\forall u \in U$ set $U^{'} = U \backslash \{u\}$ and $\mathcal{KS}^{'} = (U^{'}, C \cup \{d\}, V, f)$, we have $PREX_{U'}(C) \neq PREX_U(C)$. Based on Remark 2, we next construct an algorithm to generate an irredundant consistent IDT from a consistent IDT. The method is designed as Algorithm 1. In essence, Algorithm 1 consists of four steps, as illustrated in Figure 1. The main steps of the algorithm include: determining the set of equal elements, determining the set of maximal equivalence sets, eliminating objects, and returning the new decision table.

Algorithm 1: Determine an Irredundant Consistent IDT

Input: IDT $\mathcal{KS} = (U, C \cup \{d\}, V, f), U = \{u_1, ..., u_m\}$, $PREX_U(C)$

Output: U' is an irredundant IDT.

1: **Step 1.** Considering $r = \{r_1, ..., r_m\}, r \in U$ and $R = C \cup \{d\}$.
2: Set $\varepsilon_r = \{E_{ij}\}$ with $E_{ij} = \{a \in R : a(u_j) = a(u_i)$ or $a(u_i) = *$ or $a(u_j) = *\}$, where $1 \leq i \leq j \leq m$.
3: Set $M_U^d = \{P \in \varepsilon_r : P \neq R, d \notin P$ and $\nexists P' \in \varepsilon_r : d \notin P'$ and $P \subset P'\}$. M_U^d is the maximal equivalence set on U of d.
4: **Step 2.**
5: Set $N_0 = U = \{r_1, ..., r_m\}$
6: Calculate: $N_{i+1} = N_i \setminus \{r_{i+1}\}$ if $M_{N_i}^d = M_U^d$
7: Otherwise, $N_{i+1} = N_i$
8: $M^d_{N_i \setminus \{r_{i+1}\}}$ is the maximal equality set on $N_i \setminus \{r_{i+1}\}$ of d.
9: **Last step.** Set $U' = N_m$

Lemma 1. *Given a consistent IDT $\mathcal{KS} = (U, C \cup \{d\}, V, f)$ with $U = \{u_1, ..., u_m\}$, and $PREX_U(C)$ is the set of all reducts of $\mathcal{KS}$, then $\mathcal{KS} = (U^{'}, C \cup \{d\}, V, f)$ is an irredundant consistent IDT, where $U^{'}$ is the set of objects determined from Algorithm 1.*

Proof. Following the steps of the above algorithm by induction, we have $M_{N_m}^d \neq M_U^d$. According to Theorem 1 and Definition 8, $PREX_{N_m}(C) = PREX_U(C)$. On the other hand, according to the construction of the algorithm, we have

$$N_m \subset ... \subset N_1 \subset N_0. \tag{1}$$

It can be easily seen that if the cardinality of N_m is equal to 1 ($|N_m| = 1$) then $\mathcal{KS} = (N_m, C \cup \{d\}, V, f)$ is non-redundant (or call it irredundant). Otherwise, if $|N_m| > 1$, then $r_i \in N_m$, according to (1) we have

$$N_{i-1} \setminus \{r_i\} \subseteq N_m \setminus \{r_i\}. \tag{2}$$

We have $M^d_{N_{j-1} \setminus \{r_j\}} \neq M_U^d$ that is consistent with the way the algorithm was constructed. From 1 and 2, we have $M^d_{N_{j-1} \setminus \{r_j\}} \neq M_U^d$. Let $U_1 = N_m \setminus \{r_j\}$. According to Theorem 1 and the definition of the antikey set we have $PREX_{U_1}(C) \neq PREX_U(C)$. So $\mathcal{KS} = (N_m, C \cup \{d\}, V, f)$ is non-redundant. Lemma 1 has been proven. ∎

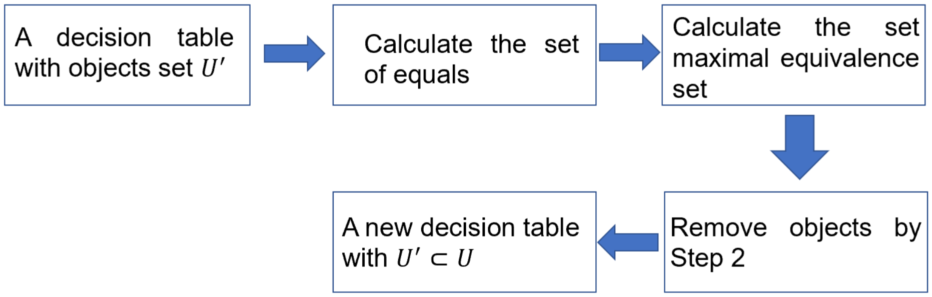

Figure 1: The processing procedure of the algorithm

Remark 3. *Two important remarks regarding the proposed algorithm.*

1) As can be observed, M_U^d is computed in polynomial time according to Algorithm 1. As a result, it is simple to show that the approach's computing complexity scales polynomial with the number of rows and columns in the $\mathcal{KS}$.

2) Permuting the order of the objects in U results in another irredundant consistent decision table.

Example 1. Given a decision table $\mathcal{KS} = (U, C \cup \{d\}, V, f)$ with five objects and C having five condition attributes, consider the following.

Table 1: The consistent IDT in Example 1

U	c_1	c_2	c_3	c_4	c_5	d
r_1	1	1	0	0	*	0
r_2	1	0	0	0	0	0
r_3	0	1	1	0	1	1
r_4	1	0	1	0	0	1
r_5	1	1	0	1	1	0

Step 1. According to the step 1 of Algorithm 1, we shall determine the maximal equivalence set on U of d:

$E_{12} = \{c_1, c_3, c_4, c_5, d\}, E_{13} = \{c_2, c_4, c_5\}, E_{14} = \{c_1, c_4, c_5\}, E_{15} = \{c_1, c_2, c_3, c_5, d\}$,

$E_{23} = \{c_4\}, E_{24} = \{c_1, c_2, c_4, c_5\}, E_{25} = \{c_1, c_3, d\}$,

$E_{34} = \{c_3, c_4, d\}, E_{35} = \{c_2, c_5\}$,

$E_{45} = \{c_1\}$.

To simplify the notation, we denote $\{c_1, c_2, \ldots, c_n\}$ by $\{c_1c_2 \ldots c_n\}$. From this, we obtain: $\varepsilon_r = \{c_1c_3c_4c_5d, c_2c_4c_5, c_1c_4c_5, c_1c_2c_3c_5d, c_4, c_1c_2c_4c_5, c_1c_3d, c_3c_4d, c_2c_5, c_1\}$. Set $\varepsilon_r = E_U$ is the set of equal sets on U. Clearly, $E_{24} \neq C \cup \{d\}$, $d \notin E_{24}$ and $\nexists P' \in \varepsilon_r$ which satisfied $d \notin P'$ and $P \subset P'$, so $M_U^d = \{c_1c_2c_4c_5\}$ is a maximal equivalence set.

Step 2. Set $N_0 = U = \{r_1, r_2, r_3, r_4, r_5\}$.

- Compute $N_0 \backslash \{r_1\} = \{r_2, r_3, r_4, r_5\}$ and $E_{N_0 \backslash \{r_1\}} = \{c_4, c_1c_2c_4c_5, c_1c_3d, c_3c_4d, c_2c_5, c_1\} \Rightarrow M^d_{N_0 \backslash \{r_1\}} = \{c_1c_2c_4c_5\} = M_U^d$. Thus, r_1 is considered a redundant object in the decision table. Therefore, $N_1 = \{r_2, r_3, r_4, r_5\}$.

- Compute $N_1 \setminus \{r_2\} = \{r_3, r_4, r_5\}$ and $E_{N_1\setminus\{r_2\}} = \{c_3c_4d, c_2c_5, c_1\} \Rightarrow M^d_{N_1\setminus\{r_2\}} \neq M^d_U$. Hence, r_2 is considered a irredundant object in the decision table. Therefore, $N_2 = N_1 = \{r_2, r_3, r_4, r_5\}$.

- Compute $N_2 \setminus \{r_3\} = \{r_2, r_4, r_5\}$ and $E_{N_2\setminus\{r_3\}} = \{c_1c_2c_4c_5, c_1c_3d, c_1\} \Rightarrow M^d_{N_2\setminus\{r_3\}} = M^d_U$. Thus, $N_3 = \{r_2, r_4, r_5\}$.

- Compute $N_3 \setminus \{r_4\} = \{r_2, r_5\}$ and $E_{N_3\setminus\{r_4\}} = \{c_1c_3d\} \Rightarrow M^d_{N_3\setminus\{r_4\}} \neq M^d_U$. Therefore, $N_4 = \{r_2, r_4, r_5\}$.

- Finally, we compute $N_4 \setminus \{r_5\} = \{r_2, r_4\}$ and $E_{N_4\setminus\{r_5\}} = \{c_1c_2c_4c_5\} \Rightarrow M^d_{N_4\setminus\{r_5\}} = M^d_U$. Thus, $N_5 = \{r_2, r_4\}$.

Step 3. Set $U' = N_5 = \{r_2, r_4\}$ and so we have $\mathcal{KS}' = (U', C \cup \{d\}, V, f)$ is an irredundant consistent IDT.

To see the effectiveness of Algorithm 1, we will next present a comprehensive example. Specifically, the decision table includes multiple missing information values. These values are present not only in the condition attributes but also in the decision attributes of the table.

Example 2. Given a decision table $\mathcal{KS} = (U, C \cup \{d\}, V, f)$ with six objects and C having six condition attributes, consider the following.

Table 2: The consistent IDT in Example 2

U	c_1	c_2	c_3	c_4	c_5	c_6	d
r_1	1	1	0	0	2	*	1
r_2	0	1	0	0	*	1	0
r_3	0	0	1	*	2	1	*
r_4	2	0	0	2	*	*	2
r_5	3	2	*	0	2	0	1
r_6	2	*	1	0	0	2	0

Step 1. Based on the step 1 of Algorithm 1, we shall determine the maximal equivalence set on U of d:

$E_{12} = \{c_2, c_3, c_4, c_5, c_6\}, E_{13} = \{c_4, c_5, c_6, d\}, E_{14} = \{c_3, c_5, c_6\}, E_{15} = \{c_3, c_4, c_5, c_6, d\}$,
$E_{16} = \{c_2, c_4, c_6\}$,
$E_{23} = \{c_1, c_4, c_5, c_6, d\}, E_{24} = \{c_3, c_5, c_6\}, E_{25} = \{c_3, c_4, c_5\}, E_{26} = \{c_2, c_3, c_4, d\}$,
$E_{34} = \{c_2, c_4, c_5, c_6, d\}, E_{35} = \{c_3, c_4, c_5, d\}, E_{36} = \{c_2, c_4, c_5, d\}$,
$E_{45} = \{c_1, c_2, c_5, c_6\}, E_{46} = \{c_2, c_4, c_5, d\}$,
$E_{56} = \{c_2, c_3, c_4\}$.

Therefore, we obtain $M^d_U = \{E_{12}, E_{45}\}$. It is easy to see that, for each $P \in M^d_U$, we can determine a different irredundant incomplete decision table from P. Therefore, Algorithm 1 can generate multiple different results.

Step 2. With $M^d_U = E_{45}$, similarly to step 2 of Example 1, we can easily obtain the set $U'_1 = \{r_4, r_5\}$ and $M^d_U = E_{12}$ we have $U'_2 = \{r_1, r_2\}$.

Step 3. From the obtained results, set $U'_1 = \{r_4, r_5\}$, we obtain the first irredundant consistent IDT $\mathcal{KS}'_1 = (U'_1, C \cup \{d\}, V, f)$ and with $U'_2 = \{r_1, r_2\}$, we obtain the second irredundant consistent IDT $\mathcal{KS}'_2 = (U'_2, C \cup \{d\}, V, f)$.

As a result, we can use the previous method to reduce the number of rows in any consistent decision table. By using the technique to identify the reduct [19], we can also reduce the columns of any consistent decision table. In other words, we can reduce the two-dimensional size of any consistent decision table by utilizing both approaches. However, the proposed method still has some drawbacks, as clearly demonstrated in Example 2 of this paper. First, it can be observed that with high-dimensional datasets containing a significant amount of missing data, the proposed algorithm may generate a large number of maximal equivalence sets. In this case, the algorithm's complexity increases when attempting to determine irredundant consistent incomplete decision tables. Second, for datasets with a wide range of values, determining maximal equivalence sets becomes infeasible. To address this issue, the initial decision table may need to undergo a data discretization process, which could result in the loss of important information. As a consequence, the final results may be affected.

4. CONCLUSIONS

Attribute reduction is one of the effective applications to improve the performance of machine learning models. In the context of big data, attribute reduction methods have played an indispensable role in the fields of knowledge discovery. However, methods for reducing objects in inconsistent decision tables are not widely known.In this study, we have generalized the basic properties of decision tables and designed an algorithm that transforms an inconsistent incomplete decision table into an irredundant incomplete decision table. Based on this result, decision tables or information systems can eliminate unnecessary objects, improving database efficiency in storage and optimizing computational space for machine learning models. Moreover, removing redundant objects in the data is also crucial. Intuitively, these objects may be generated due to noise from data entry errors or some sensor faults during the measurement process. Storing these objects can significantly impact the performance of machine learning models. Therefore, this study can be considered a promising tool that can be applied to knowledge discovery technologies. Hence, it can be asserted that the irredundant decision table is a highly effective and versatile tool for representing information.

REFERENCES

[1] Z. Pawlak, *Rough Sets: Theoretical Aspects of Reasoning About Data.* Norwell, MA: Kluwer Academic, 1991.

[2] M. Kryszkiewicz, "Rough set approach to incomplete information systems," *Information Sciences*, vol. 112, no. 1-4, pp. 39–49, 1998.

[3] D. T. Khanh, V. D. Thi, N. L. Giang, and L. H. Son, "Some problems related to reducts of consistent incomplete decision tables," *International Journal of Mathematical, Engineering and Management Sciences*, vol. 7, no. 2, pp. 288–298, 2022.

[4] J. Demetrovics, D. Vu, and L. Nguyen, "An efficient algorithm for determining the set of all reductive attributes in incomplete decision tables," *Cybernetics and Information Technologies*, vol. 13, no. 4, pp. 118–126, 2013.

[5] V. D. Thi and N. L. Giang, "A method for extracting knowledge from decision tables in terms of functional dependencies," *Cybernetics and Information Technologies*, vol. 13, no. 1, pp. 73–82, 2013.

[6] V. Thi and N. L. Giang, "A method to construct decision table from relation scheme," *Cybernetics and Information Technologies*, vol. 11, no. 3, pp. 32–41, 2011.

[7] N. L. Giang and V. D. Thi, "Some problems concerning condition attributes and reducts in decision tables," in *Proceeding of the Fifth National Symposium "Fundamental and Applied Information Technology Research" (FAIR)*, Dong Nai, Viet Nam, 2011, pp. 142–152.

[8] P. V. Anh, V. D. Thi, and N. N. Cuong, "A novel algorithm for finding all reducts in the incomplete decision table," *Journal of Computer Science and Cybernetics*, pp. 313–321, 2023.

[9] J. Demetrovics, H. M. Quang, V. D. Thi, and N. V. Anh, "An efficient method to reduce the size of consistent decision tables," *Acta Cybernetica*, vol. 23, no. 4, pp. 1039–1054, 2018.

[10] J. Demetrovics, V. D. Thi, and N. L. Giang, "On finding all reducts of consistent decision tables," *Cybernetics and Information Technologies*, vol. 14, no. 4, pp. 3–10, 2014.

[11] N. L. Giang, J. Demetrovics, V. D. Thi, and P. D. Khoa, "Some properties related to reduct of consistent decision systems," *Cybernetics and Information Technologies*, vol. 21, no. 2, pp. 3–9, 2021.

[12] J. Demetrovics, "On the equivalence of candidate keys with sperner systems," *Acta Cybernetica*, vol. 4, no. 3, pp. 247–252, 1979.

[13] J. Demetrovics and V. D. Thi, "Algorithms for generating an armstrong relation and inferring functional dependencies in the relational datamodel," *Computers & Mathematics with Applications*, vol. 26, no. 4, pp. 43–55, 1993.

[14] J. Demetrovics and V. Thi, "Keys, antikeys and prime attributes," *Annales Univ. Sci. Budapest, Sect. Comp.*, vol. 8, 1987.

[15] J. Demetrovics and V. D. Thi, "Relations and minimal keys," *Acta Cybernetica*, vol. 8, no. 3, pp. 279–285, 1988.

[16] J. Demetrovics and T. V.D, "Some remarks on generating armstron and inferring functional dependencies relation," *Acta Cybernetica*, vol. 12, no. 2, pp. 167–180, 1995.

[17] J. Demetrovics and V. D. Thi, "Some results about functional dependencies," *Acta Cybernetica*, vol. 8, no. 3, pp. 273–278, 1988.

[18] V. D. Thi, "Minimal keys and-antikeys," *Acta Cybernetica*, vol. 7, no. 4, pp. 361–371, 1986.

[19] J. Demetrovics, N. L. Giang, V. D. Thi, and P. V. Anh, "On the problem related to reductive attributes in the incomplete decision tables," *Serdica Journal of Computing*, vol. 16, no. 1, pp. 24–38, 2022.

Received on December 28, 2024
Accepted on March 08, 2025