FINDING KNOWLEDGE ACCORDING TO ROUGH SET THEORY

Phung Thi Thu Hien*, Ninh Van Tho
University of Economic and Technical Industries

ABSTRACT

Attribute reduction is a core issue of rough set theory and also an essential pre-processing step in data mining. In recent years, there have been many papers about attribute reduction methods based on different views, and generally can be classified as attribute reduction method based on positive region, attribute reduction method based on discernibility matrix, attribute reduction method used information entropy. However, most of attribute reduction methods are performed on single-valued decision system decision table. In this paper, we propose methods for attribute reduction in set-valued decision systems. Next, based on some results in the relational database, this article proposes an algorithm building a relationship scheme from the decision table.

Keywords: *Relational database, rough set, relational scheme, decision tablem, keys*

INTRODUCTION

The theory of conventional rough set initiated by Pawlak [4] is an effective tool to solve attribute reduction problems and to extract rules in information systems. Attribute reduction in decision systems is the process of choosing the minimum set of the conditional attribute set, preserving classified information of the decision systems. In decision systems, computer scientists have provided several attribute reduction methods based on model of conventional rough set, summarized by Shifei D et. al. in ref. [10]. In set-valued information system, Guan Y. Y. Wang et. al. [6] expanded equivalent relation in conventional rough set to tolerance relation and developed model tolerance-based rough set by expanding lower approximation, upper approximation, positive domain, etc. based on tolerance relation. There are remarkable reports about attribute reduction in decision system and ordered decision system in model of tolerance-based rough set approach in ref. [2], [9], [13]. In ref. [15], the authors using matrix method studied the altering of approximation sets with and without attribute set.

In this paper, section 2 describes the results of set-valued decision system and definitions of reduct and basic concepts in relational databases. In section 3, the author demonstrate attribute reduction method. In section 4, the author provides some algorithms in relation database. In section 5, the author discuss about the overall results and future study.

BASIC DEFINITIONS

Basic definitions in rough set

A decision table is defined as $DT=(U,C\cup\{d\})$ in which $U=\{u_1,u_2,...,u_n\}$ is the finite & non-empty set of objects $C=\{c_1,c_2,...,c_m\}$ the set of condition attributes, D is the set of decision attributes and $C\cap D=\varnothing$, $V=\prod_{a\in C\cup D} V_a$ where V_a is the value range of attribute a, $f:U\times(C\cup D)\rightarrow V$ is an information function, where $\forall a\in C\cup D, u\in U$, $f(u,a)\in V_a$ hold.

Set-valued decision systems were proposed as a tool to characterize the data sets with incomplete or uncertain information [9].

Formally set-values decision table is a tuple $DT=(U,A\cup\{d\})$, where U is a finite set of objects, A is a finite set of set-valued attributes, i.e the functions of form $a:U\rightarrow 2^{V_a}$ for $a\in A$, and $d\notin A$ is a distinguished attribute called decision. The set V_a is called the domein of attribute a, and $a(x)\subseteq V_a$ for each $a\in A$ and $x\in U$. In the case, when $|a(x)|=1$ for any $a\in A$ and $x\in U$ we have a standard single valued decision table.

In Table 1 [9] we have an example of a set-valued decision system.

* Tel: 0914 770070, Email: Thuhiencn1@gmail.com

Table 1. *An example of a set-valued decision table*

U	Audition(**A**)	Spoken Language(**S**)	Reading(**R**)	Writing(**W**)	dec
x_1	$\{E\}$	$\{E\}$	$\{F,G\}$	$\{F,G\}$	No
x_2	$\{E,F,G\}$	$\{E,F,G\}$	$\{F,G\}$	$\{E,F,G\}$	No
x_3	$\{E,G\}$	$\{E,F\}$	$\{F,G\}$	$\{F,G\}$	No
x_4	$\{E,F\}$	$\{E,G\}$	$\{F,G\}$	$\{F\}$	No
x_5	$\{F,G\}$	$\{F,G\}$	$\{F,G\}$	$\{F\}$	No
x_6	$\{F\}$	$\{F\}$	$\{E,F\}$	$\{E,F\}$	Yes
x_7	$\{E,F,G\}$	$\{E,F,G\}$	$\{E,G\}$	$\{E,F,G\}$	Yes
x_8	$\{E,F\}$	$\{F,G\}$	$\{E,F,G\}$	$\{E,G\}$	Yes
x_9	$\{F,G\}$	$\{G\}$	$\{F,G\}$	$\{F,G\}$	Yes
x_{10}	$\{E,F\}$	$\{E,G\}$	$\{F,G\}$	$\{E,F\}$	Yes

Let $DT=(U, A\cup\{d\})$ be a set-valued decision table. Any reflexive and symmetric relation T ⊆ U × U is called a tolerance relation defined on U. A tolerance relation T_B related to a set of attributes B ⊆ A can be defined by:

$T_B(x, y) \Leftrightarrow \forall b\in B\ |a(x) \cap a(y)| \neq \emptyset$ (1)

For any $B\subseteq A$ we denote by $[x]_{T_B}=\{y\in U:(x,y)\in T_B\}$ the tolerance class related to object x ∈ U. We also denote by the family $U/T_B=\{[x]_{T_B}: x\in U\}$ of all tolerance classes of T_B .

Basic concepts in relational databases [1],[4], [12]. Let $R=\{a_1,...,a_n\}$ be a nonempty finite set of atributes, each attribute has a domain value of $D(a_i)$. A relation r on R as a set of tuples $\{h_1,...,h_m\}$, $h_j: R\to \bigcup_{a_i\in R} D(a_i)$, $1\le j\le m$ is a function such that $h_j(a_i)\in D(a_i)$.

Let $r=\{h_1,...,h_m\}$ be a relation over $R=\{a_1,...,a_n\}$. A functional dependency (*FD* for short) over *R* is a statement of form $A\to B$, where A, B $\subseteq$ R. *FD* $A\to B$ holds in a relation *r* over *R* if

$$(\forall h_i,h_j\in r)\left(\begin{array}{l}(\forall a\in A)(h_i(a)=h_j(a))\Rightarrow\\(\forall b\in B)(h_i(b)=h_j(b))\end{array}\right)$$

Let $F_r=\{(A,B): A,B\subseteq R, A\to B\}$, F_r is called the full family of functional dependencies in *r*. Let *R* be a finite set and denote *P(R)* its power set, we say that *F* is an *f*-family over *R* iif for all $A,B,C,D\subseteq R$:

(1) $(A, A)\in F$
(2) $(A, B)\in F, (B, C)\in F \Rightarrow (A, C)\in F$
(3) $(A, B)\in F, A\subseteq C, D\subseteq B \Rightarrow (C, D)\in F$
(4) $(A, B)\in F, (C, D)\in F \Rightarrow (A\cup C, B\cup D)\in F$

Clearly, F_r is an *f*-family over *R*. It is known [1] that if *F* is an arbitrary *f*-family over *R*, then there is a relation *r* such that $F_r=F$. F^+ is the set of all *FDs* which can be derived from *F* by the rules $(1)-(4)$.

A relation schema *s* is a pair $<R,F>$, where *R* is a set of attributes and *F* is a set of *FDs* on *R*. Denote $A^+=\{a\in R \mid A\to\{a\}\in F^+\}$, A^+ is called the *closure* of *A* on *s*.

It is clear that $A\to B\in F^+$ iif $B\subseteq A^+$. According to [1], if *s* = <*R*, *F*> is a relational schemes *r* over R, such a relation is called an Armstrong relation of *s*.

Let *r* be a relation, $s=<R,F>$ be a relation scheme and $A\subseteq R$. Then *A* is a key of *r* (a key of *s*) if $A\to R\,(A\to R\in F^+)$. *A* is a minimal key of *r* (*s*) if *A* is a key of *r* (*s*) and any proper subset of *A* is not a key of *r* (*s*). Denote $K_r\,(K_s)$ the set of all minimal keys of *r* (*s*). $K\subseteq P(R)$ is a Sperner system if for any $K_1,K_2\in K$ implies $K_1\not\subset K_2$. Clearly, $K_r\,(K_s)$ are Sperner systems.

Let *K* be a Sperner-system over *R* as the set of all minimal keys of *s* . We defined the set of antikeys of *K*, denoted by K^{-1}, as follows:

$K^{-1}=\{A\subset R:(B\in K)\Rightarrow(B\not\subset A)\}$ and if

$(A\subset C)\Rightarrow(\exists B\in K)(B\subseteq C)$.

It is easy to see that K^{-1} is also a Sperner system over *R*. By definition, if *K* is the minimum set of keys of a *FD* then K^{-1} is the set of all set not the biggest key.

Let *r* be a relation over *R*. Denote $E_r=\{E_{ij}: 1\le i<j\le|r|\}$, where $E_{ij}=\{a\in R: h_i(a)=h_j(a)\}$. Then E_r is called the equality set of *r*. It is known [2] that for $A_r\subseteq R$, $A_r^+=\cap E_{ij}$ if there exists $E_{ij}\in E_r: A\subseteq E_{ij}$, otherwise $A_r^+=R$. In next content, we introduce some definitions about the family of all minimal sets of an attribute over a relation and a relation scheme.

Definition 2.[4] Let $s=(R,F)$ be a relation scheme over *R* and $a\in R$.

Set

$K_a^s=\{A\subseteq R: A\to\{a\}, \nexists B:(B\to\{a\})(B\subset A)\}$.

K_a^s is called the family of minimal sets of the attribute *a* over *s*.

Similarly, we define the family of minimal sets of an attribute over a relation

Definition 3. Let *r* be a relation over *R* and $a\in R$.

Set

$K_a^r=\{A\subseteq R: A\to\{a\}, \nexists B\subseteq R:(B\to\{a\})(B\subset A)\}$

K_a^r is called the family of minimal sets of the attribute *a* over *r*. It is clear that $R\notin K_a^s, R\notin K_a^r, \{a\}\in K_a^s, \{a\}\in K_a^r$ and K_a^s, K_a^r are Sperner systems over *R*.

ATTRIBUTE REDUCTION IN SET-VALUED DECISION SYSTEM

Attribute reduction in decision systems is the process of choosing the minimum set of the conditional attribute set, preserving classified information of the decision syste

Definition 4. (Decision relative reduct)

Given a set-valued decision table $DT=(U,A\cup\{d\})$ the decision relative reduct of DT is the minimal set of attribute R ⊆ A, which satisfying the following conditions:

1. for any pair (x, y) ∈ U, if d(x) ≠ d(y) and (x, y) ∉ T_A then (x, y) ∉ T_R;

2. no proper subset R'of R satisfies the previous condition.

The reduct R is optimal if it consists of the smallest number of attributes.

Discernibility Function

Definition 5. (Basic discernibility measure) [11]

Let $DT=(U,A\cup\{d\})$ be a single-valued decision table. The discernibility measure for a set of attributes B ⊆ A is defined by:

$$disc(B)=\left|\{(x,y)\in U\times U\,|\,(d(x)\neq d(y))\wedge\exists_{b\in B}(b(x)\neq b(y))\}\right|$$

Definition 6. (Generalized discernibility function). Let $DT=(U,A\cup\{d\})$ be a set-valued decision table with tolerance relations T_a (for all a ∈ A). The mapping discern : 2^A: $R^+\cup\{0\}$, defined by

$$discern(B)=\left|\left\{\begin{array}{l}(x,y)\in U\times U\,|\,(d(x)\neq d(y))\wedge\\ \exists_{b\in B}(x,y)\in T_b\end{array}\right\}\right|$$

where B ⊆ A is set of attributes, is called the generalized discernibility function.

Below we list some properties of the generalized function:

Property 1. For any attribute a ∈ A, the value discern(a) is equal to frequency of occurrence of attribute a in the discernibility matrix M_{DT}.

Property 2. Discernibility function is increasing. For any set B ⊆ A and C ⊆ A, if B ⊆ C then discern(B) ≤ discern(C).

Contingency Table and Tolerance-Based Contingency Table

Contingency Table.

Let V_d be the set of decision values in decision table $DT=(U,A\cup\{d\})$, and let $U/IND(B)=\{[x_1]_B,[x_2]_B,....,[x_{n_S}]_B\}$ be partition of U defined by indiscernibility relation IND(B) for $B\subseteq A$. Contingency table CT_B related to B is a two dimensional table $CT_B=[CT_B[i,j]]_{i\in\{1,...,n_B\}}^{j\in\{1,...,|V_d|\}}$

where: $CT_B[i,j]=|\{x\in U: x\in[x_i]_B\wedge d(x)=j\}|$

The local discernibility measure related to indiscernibility class $[x_i]_B$ is defined as follows:

$$\delta([x_i]_B)=\left|\{(x_1,x_2)\in[x_i]_B\times(U\setminus[x_i]_B: d(x_1)\neq d(x_2)\}\right|$$

$$=\sum_{j_1\neq j_2, u_k\notin[u_i]_B} CT[i,j_1].CT[k,j_2]$$

$$=\sum_{j_1\neq j_2} CT[i,j_1].(|D_{j_2}|)-CT[i,j_2]$$

where $||D_j|$ denotes cardinality of decision class D_j for $j=1,...|V_d|$

Hence the basic discernibility measure of attribute set B is defined as the number of pairs of discernible objects, i.e.

$$disc(B)=\sum_i \delta_B([x_i]_B)=\frac{1}{2}\sum_{i=1}^{n_B}\sum_{j_1\neq j_2} CT[i,j_1].CT[i,j_2]) \quad (2)$$

Table 2. *The contingency tables for single attributes and values of the discern function of spoken language attribute*

Spoken language		
Values	**No**	**Yes**
E	1	0
F	0	1
G	0	1
E,F	1	0
E,G	1	1
F,G	1	1
E,F,G	1	1
discern (S) = 22		

The summation is taken over the disjoint subsets induced by IND(B) and over all $j_1, j_2 \in \{1, \ldots |V_d|\}, j_1 \neq j_2$.

Table 2 presents the contingency table and the values of the discernibility function for each attribute from Table 1. We remind that the cardinality of each decision class is equal to 5. The contingency table with the indiscernibility relation is further called the basic contingency table.

Proposition 1. Let $DT = (U, A \cup \{d\})$ be a decision table. Let IND(B) be a indiscernibility relation related to $B \subseteq A$. Let n_B denotes a number of indiscernibility classes defined by INB(B). Given a contingency table CT_B. The value discern(B) can be determined in time O(dn_B), which is bounded by O(dn), where n = |U | and d is a number of decision classes.

Tolerance-Based Contingency Table. For a decision table $DT = (U, A \cup \{d\})$, let T_B be a tolerance relation for $B \subseteq A$ and let $U / IND(B) = \{[x_1]_B, [x_2]_B, \ldots, [x_{n_s}]_B\}$ be the partition of U defined by indiscernibility relation IND(B). The tolerance based contingency table is a two-dimensional table $TCT_B = [TCT[i,j]]_{i \in \{1,\ldots n_B\}}^{j \in \{1,\ldots |V_d|\}}$, which is defined as follows:

$$TCT_B[i,j] = \left|\{u \in U \mid u \in [u_i]_B \ \text{và} \ d(u) = j\}\right|$$

Intuitively, tolerance-based contingency table stores the decision distributions inside each tolerance class. One can observe that the tolerance classes are not disjoint in general. To compute the value of discernibility function we modify the concept of a local discernibility measure.

For a tolerance class $[x_i]_{T_B}$, the local discernibility measure related to $[x_i]_{T_B}$ is defined by:

$$\delta_B([x_i]_{T_B}) = |\{(x_1, x_2) \in [u_i]_{T_B} \times (U \setminus [x_i]_{T_B}) : d(x_1) \neq d(x_2)\}|$$

$$= \sum_{j_1 \neq j_2, x_k \notin [x_i]_{T_B}} CT_B[i, j_1] \times CT_B[k, j_2]$$

$$= \sum_{j_1 \neq j_2} CT_B[i, j_1](|D_{j_2}| - TCT_B[i, j_2])$$

The generalized discernibility measure can be calculated as follows:

$$Discern(B) = \sum_i \delta_B([x_i]_{T_A}) = \frac{1}{2} \sum_{i=1}^{n_A} \sum_{j_1 \neq j_2} CT_B[i, j_1](|D_{j_2}| - TCT_B[i, j_2]) \quad (3)$$

where $B \subset A$. We denote by $CT_B \otimes TCT_B$ the operation in Equation 3. The summation is taken over a disjoint subsets induced by IND(B) and over all $j_1, j_2 \in \{1, \ldots |V_d|\}, j_1 \neq j_2$.

Algorithm attribute reduction in set-valued decision tables

Algorithm 1. Generalized Maximal Discernibility heuristic for setvalued decision tables with tolerance relation.

1: **Input**: Set-valued decision table D = (U, A ∪ d).

2: **Output**: Attribute reduction R.

3: Generate a set of lattices Latt(A);

4: R ← ∅;

5: discern(R) ← 0;

6: while (discern(R) < discern(A)) do

7: max discern ← 0;

8: for (ai ∈ A) do

9: B ← R ∪ {ai };

10: Create CT_B ;

11: Create TCT_B using CT_B;

12: Determine discern(B) = $CT_B \otimes TCT_B$ using Equation (3);

13: if (discern(B) > max dicern) then

14: max discern ← discern(B);

```
    best attribute ← ai ;
  end if
 end for
 A ← A \ {best attribute};
 R ← R ∪ {best attribute};
end while
```

The time complexity of Algorithm 3.3 is $O(k^3n^2)$, where k is a number of attributes, n is the number of objects.

BASIC ALGORITHMS IN RELATION DATABASE

Finding a minimal key is one of the most important problems in the field of knowledge discovery and data mining.

Algorithm 2. [3] Finding a minimal key from the set of antikeys.

Input: Let *K* be a Sperner-system over *R* as the set of antikeys, $C=\{b_1,...,b_m\}\subseteq R$ and *H* is a Sperner-system as the set of minimal keys $\left(K=H^{-1}\right)$ such that $\exists B\in K: B\subseteq C$

Output: $D\in H$

Step 1: We set T(0) = C;

Step i+1: We set

$T(i+1)=T(i)-b_{i+1}$ if $\forall B\in K$, there is not $T\subseteq B$

$T(i+1)=T(i)$ otherwise

Finally, we set *D = T(m)*;

Algorithm 3. [3] Finding the set of minimal keys from the set antikeys.

Input: Let $K=\{B_1,...,B_k\}$ be a Sperner-system over *R*.

Output: *H* where $H^{-1}=K$

We construct *H* by induction.

Step 1: We construct an $A_1, (A_1\in H)$ using Algorithm 2 We set $K_1=A_1$.

Step i+1: If there is a $B\in K_i^{-1}$ such that $B\not\subseteq B_j\ (\forall j:1\le j\le m)$,then by algorithm which finds a minimal key (Algorithm 2) we determine an A_{i+1}, where $A_{i+1}\in H, A_{i+1}\subseteq B$. After that, let $K_{i+1}=K_i\cup A_{i+1}$. In the converse case we set $H=K_i$.

From definition 3, the article builds the algorithm for finding the minimal set of attributes over relation.

Algorithm 4. Algorithm finds the minimal set of attributes over relation

Input: $r=\{u_1,u_2,...,u_m\}$ is the relation over *R* and $a\in R$.

Output: K_a^r.

Step 1: From *r* we calculate the equality system $E_r=\left\{E_{ij}:1\le i<j\le m\right\}$, where $E_{ij}=\left\{a\in R: u_i(a)=u_j(a)\right\}$.

Step 2: From E_r we construct the set $M_a=\left\{A\in E_r: a\notin A \nexists B\in E_r: a\notin B, A\subset B\right\}$.

Step 3: Compute *K* from the set M_a $\left(K^{-1}=M_a\right)$(By Algorithm 3.)

In the worst case, the complexity of the algorithm is not greater than the exponent n in which n is the number of elements of R.

Algorithms to construct relation scheme from decision table

The problem: Given a decision table $DS=\left(U,C\cup\{d\}\right)$ as a relation *r* over an attribute $R=C\cup\{d\}$, we have to construct the relation scheme $s_d=<R,F>$, where F is the set of functional dependencies $A_i\to\{d\}$ for $A_i\subseteq C, 1\le i\le t$, such that $K_d^s=K_d^r=RED(C)\cup\{d\}$, where K_d^s is the set of all minimal keys of s_d, K_d^r is the family of all minimal sets of the attribute *d* over the relation *r* and *RED(C)* is the set of all reducts of *DS*.

Algorithm 5. Construct a relation scheme from a decision table.

Input: Let $DS=\left(U,C\cup\{d\}\right)$ be a decision table, where $POS_C\left(\{d\}\right)=U$.

Output: $s_d=<R,F>$ such that $K_d^s-\{d\}=R_d$.

Let us consider the relation *r* over the set of attributes $R = C \cup \{d\}$.

Step 1: Using Algorithm 3 we obtain K_d^r. Assume that $K_d^r = \{K_1, K_2, ..., K_t\}$, according to definition K_d^r is a Sperner-system over *C*.

Step 2: For each $K_i \in K_d^r, 1 \le i \le t, K_i \ne \{d\}$, we construct the functional dependency $K_i \to \{d\}$. The relation scheme $s_d = < R, F >$, where $R = C \cup \{d\}$ and $F = \{K_i \to \{d\} : K_i \in K_d^r\}$, is the one we have to construct.

The complexity of the algorithm is polynomial according to the size of r.

Proof $K_d^s - \{d\} = R_d$ *first of all, I prove* $K_d^s = K_d^r$

1) For any $K \in K_d^r$ we have $K \to \{d\}$ and there does not exist $K' \subset K$ such that $K' \to \{d\}$. Hence, according to the method to construct $s_d = < R, F >$ we conclude *K* is a minimal key of s_d, that is $K \in K_d^s$.

2) Conversely, assume that there exists $K \in K_d^s$ such that $K \notin K_d^r$, then we have $K \to \{d\}$ and there does not exist $K' \subset K$ such that $K' \to \{d\}$. It is easy to see that for any $K_i \in K_d^r, 1 \le i \le t$, $K \not\subset K_i$ *(i)* because if $K \subset K_i$ then K_i is not a reduct of *C* in *DS*. Moreover, for any $K_i \in K_d^r, 1 \le i \le t, K_i \not\subset K$ *(ii)* because if $K_i \subset K$ then *K* is not a minimal key of K_d^s. From *(i)*, *(ii)* we can conclude $\mathcal{K} = \{K, K_1, K_2, ..., K_t\}$ is a Sperner-system and for any $A \subset \mathcal{K}$ we have $A \to \{d\}$. According to the definition, $\mathcal{K}$ is the family of all minimal sets of attribute *d*, so $\mathcal{K} = K_d^r, K \in K_d^r$. This is in contradiction with the condition $K \notin K_d^r$. Therefore we have $K \in K_d^r$. From 1) and 2) we conclude $K_d^s = K_d^r$.

CONCLUSION

In this paper, based on indiscernibility matrix and indiscernibility function in traditional rough set theory [11], the author proposed contingency tables and discernibility function in order to find reduct of set-valued decision system. Based on some results of J. Demetrovics and Thi V.D concerning keys,, the article building algorithm relation scheme from a consistent decision table, it has important implications in knowledge discovery and data mining. In next papers we will show that the proposed solution can be also modified to manage with dominance based rough sets approach to set-valued decision table.

REFERENCES

1. Armstrong W. W. (1974), "Dependency structures of database relationships", *Information Processing*, 74, 580-583.
2. Demetrovics J., Thi V. D. (1987), "Keys, antikeys and prime attributes". *Ann. Univ. Scien. Budapest Sect. Comput.*, 8, pp. 37-54
3. Demetrovics J., Thi V. D. (1998), "Relations and minimal keys", *Acta Cybernetica* 8, 3, pp. 279-285.
4. Demetrovics J., Thi V. D. (1995), "Some remarks on generating Armstrong and inferring functional dependencies relation", *Acta Cybernetica* 12, pp. 167-180.
5. Guan Y. Y., Wang H. K., (2006), "Set-valued information systems", *Information Sciences*, 176, pp. 2507–2525.
6. Kryszkiewicz M., (1998), "Rough set approach to incomplete information systems", *Information Science*, Vol. 112, pp. 39-49.
7. Pawlak Z., (1982), *Rough sets, International Journal of Information and Computer Sciences*, 11(5), pp. 341-356.
8. Pawlak Z. (1991), *Rough sets: Theoretical Aspects of Reasoning About Data*, Kluwer Academic Publishers.
9. Qian Y. H., Dang C. Y., Liang J. Y., Tang D. W. (2009), "Set-valued ordered information systems", *Information Sciences*, 179, pp. 2809-2832.
10. Shifei D., Hao D. (2010), "Research and Development of Attribute Reduction Algorithm Based on Rough Set", *IEEE*, CCDC2010, pp.648-653.
11. Skowron A., Rauszer C. (1992), "The Discernibility Matrices and Functions in Information systems, Interlligent Decision Support, Handbook of Applications and Advances

of the Rough Sets Theory”, *Kluwer, Dordrecht,* pp. 331-362.

12. Thi V.D, (1986), “Minimal keys and Antikeys”. *Acta Cybernetica* 7, 4 361-371.

13. Y. H. Qian Y. H. , Liang J. Y.,(2010), “On Dominance Relations in Disjunctive Set-Valued Ordered Information Systems”, *International Journal of Information Technology & Decision Making* Vol. 9, No. 1, pp. 9–33.

14. Yao Y. Y., Zhao Y., Wang J., (2006), “On reduct construction algorithms”, *Proceedings of International Conference on Rough Sets and Knowledge Technology,* pp. 297-304.

15. Zhang J. B., Li T. R., Ruan D., Liu D. (2012), “Rough sets based matrix approaches with dynamic attribute variation in set-valued information systems”, *International Journal of Approximate Reasoning* 53, pp. 620–635.

TÓM TẮT

PHÁT HIỆN TRI THỨC THEO HƯỚNG TIẾP CẬN TẬP THÔ

Phùng Thị Thu Hiền*, Ninh Văn Thọ

Trường Đại học Kinh tế Kỹ thuật Công nghiệp

Rút gọn thuộc tính là bài toán quan trọng nhất trong lý thuyết tập thô. Trong những năm gần đây, các phương pháp rút gọn thuộc tính đã thu hút sự chú ý và quan tâm của nhiều nhà nghiên cứu. Đáng chú ý là phương pháp dựa trên miền dương, phương pháp sử dụng ma trận phân biệt, phương pháp sử dụng entropy thông tin ...vv. Tuy nhiên, hầu hết các phương pháp này đều thực hiện trên các hệ thông tin đơn trị. Trong bài báo này, tác giả đưa ra phương pháp rút gọn thuộc tính trong bảng quyết định đa trị. Đồng thời, dựa trên một số kết quả nghiên cứu trong cơ sở dữ liệu quan hệ bài báo trình bày thuật toán xây dựng sơ đồ quan hệ từ bảng quyết định đơn trị.

Từ khóa: *Cơ sở dữ liệu quan hệ, tập thô, sơ đồ quan hệ, bảng quyết định, khóa.*

Ngày nhận bài: 30/8/2017; Ngày phản biện: 08/9/2017; Ngày duyệt đăng: 30/11/2017

* *Tel: 0914 770070, Email: Thuhiencn1@gmail.com*