International Journal of Approximate Reasoning 82 (2017) 56–80

Contents lists available at ScienceDirect

International Journal of Approximate Reasoning

www.elsevier.com/locate/ijar

Quick general reduction algorithms for inconsistent decision tables

Ge Hao [a,b], Li Longshu [b,c,*], Xu Yi [b,c], Yang Chuanjian [d]

[a] School of Electronic and Electrical Engineering, Chuzhou University, Chuzhou 239000, People's Republic of China
[b] Key Laboratory of Computation Intelligence and Signal Processing of Education Ministry, Anhui University, Hefei 230601, People's Republic of China
[c] School of Computer Science and Technology, Anhui University, Hefei 230601, People's Republic of China
[d] School of Computer and Information Engineering, Chuzhou University, Chuzhou 239000, People's Republic of China

ARTICLE INFO

Article history:
Received 2 August 2015
Received in revised form 25 November 2016
Accepted 26 November 2016
Available online 2 December 2016

Keywords:
Rough set
Attribute reduction
Discernibility matrix
Relative discernibility

ABSTRACT

In the Pawlak rough set model, attribute reduction plays one of the important roles, and the preservation of different properties of the original decision table leads to different types of reduct definitions, such as relative relation reduct, positive-region reduct, distribution reduct, maximum distribution reduct, and assignment reduct. However, there are no general quick reduction approaches for obtaining various types of reducts; this motivated us to conduct the present study. We first establish a unified decision table model for five representative reducts in inconsistent decision tables, study the relative discernibility and relative discernibility reduct for the general decision table, and derive the corresponding properties. Then, two general reduction algorithms (GARA-FS$_\triangle$ and GARA-BS$_\triangle$) from the viewpoint of the relative discernibility in inconsistent decision tables are presented. Subsequently, to increase the efficiency of algorithms, two quick general reduction algorithms (QGARA-FS$_\triangle$ and QGARA-BS$_\triangle$) are proposed mainly by reducing the sort times to increase the efficiency of reduction algorithms. Finally, a series of experiments with UCI data sets are conducted to evaluate the effectiveness and performance of the proposed reduction algorithms.

1. Introduction

Rough set theory was introduced by Polish mathematician Pawlak [24,25] in 1982, which is a valid mathematical theory to handle imprecise, uncertain, and vague information. It has been widely applied in many fields such as machine learning [33], data mining [15], intelligent data analyzing [4], and control algorithm acquisition [34]. Some of the features in huge-volume and high-dimensional data sets are irrelevant or redundant, which typically deteriorates the performance of machine-learning algorithms [29]. Attribute reduction is one of the key topics in the rough set theory, which can find a subset of attributes that provides the same description, discernibility, or classification ability as the original conditional attribute set. In general, attribute reduction uses a preprocessing procedure to reduce the complexity of data mining or knowledge discovery. In recent years, attribute reduction has drawn wide attention, and many researchers have focused on two aspects: various types of reduct definitions based on different criteria and quick attribute reduction algorithms.

* Corresponding author.
E-mail addresses: togehao@126.com (H. Ge), Lilongshu_ahu@126.com (L. Li), xuyi1023@126.com (Y. Xu), tocjy474@126.com (C. Yang).

http://dx.doi.org/10.1016/j.ijar.2016.11.016
0888-613X/

The different purposes for preserving a certain property embedded in original decision table will result in different notions and results of reduct. The positive-region reduct proposed by Pawlak is the classical concept of reducts [24,25], which focuses on preserving the positive region unchanged. Kryszkiewicz presented five notions of reducts for inconsistent decision tables. In fact, there only existed two types of reducts: the assignment reduct and the distribution reduct [8,9]. The assignment reduct maintains unchanged property of the possible decisions for an arbitrary object in an inconsistent decision table. In comparison, the distribution reduct is a more complete knowledge reduct that is characterized by preserving the class membership distribution for all the objects in an inconsistent decision table. In other words, the distribution reduction preserves not only all the deterministic information, but also the nondeterministic information of an inconsistent decision table. On the basis of Kryszkiewicz's work, Zhang et al. proposed the maximum distribution reduct [41,42]. It maintains unchanged property of the maximum decision classes for all the objects in a decision table, which is seen as a good compromise between the capability of preserving information with respect to decisions and the compactness of the derived rules. In refs. [41] and [42], Zhang et al. provided discernibility matrix-based reduction methods for distribution reduct, assignment reduct, and maximum distribution reduct. Ye et al. [40] presented M-reduct, which has stricter requirements than the maximum distribution reduct, in which M-reduct rigidly retains the membership degree to the maximum decision class for each object of the decision table. This characteristic makes M-reduct more susceptible to noise-contaminated data sets than the maximum distribution reduction. Miao et al. [20] presented three types of reducts for inconsistent decision tables: the region preservation reduct, the decision preservation reduct, and the relationship preservation reduct, and common definitions of relative reduct and discernibility matrices for three reducts were discussed. Liu et al. [17] presented a method to convert these types of reducts (the distribution reduct, the maximum distribution reduct, and the generalized decision reduct) into traditional reducts, and designed an efficient algorithm for the traditional reduct. Zhou et al. [45] proposed 13 typical forms of objective functions for attribute reduction in the complete decision table. However, among the 13 types of reducts for the decision table, it has been proved that there are only six and two intrinsically different reducts for inconsistent and consistent decision tables, respectively. Meng et al. [23] have recently presented a relative systematic study of attribute reduction in inconsistent incomplete decision tables, where five types of discernibility function-based approaches are proposed to identify a specific type of reducts.

The other research of attribute reduction with rough set mainly concentrates on developing feature selection algorithms to find reducts according to different models of rough sets [1–3,10,22,32,34,35], and these methods normally can be grouped into two classes: discernibility matrix-based reduction method and heuristic reduction method. Although discernibility matrix-based method can find all of the reducts, the conversion from conjunction normal form to disjunction normal form constitutes a nondeterministic polynomial (NP)-hardness problem. When data sets have many attributes and objects, discernibility matrix-based method will become nonfeasible, as the matrix contains too many candidates. The heuristic reduction method is one of the desirable methods for overcoming the drawbacks of the discernibility matrix approaches. Shen et al. [28–30] presented QuickReduct algorithm, which started with empty subset, and then added the most significant feature into the candidate set of reduct until dependency reached the maximum value in the data set. Miao et al. [21] introduced an information representation to the rough set theory and proposed a heuristic algorithm based on mutual information for attribute reduction. Liu et al. [16] proposed a new conditional information entropy. By using this entropy, the new significance of an attribute was defined, and an efficient heuristic algorithm for computing knowledge reduct was designed. Yang et al. [38] proposed a heuristic feature selection algorithm in incomplete decision tables, which keeps the positive region of target decision unchanged. Liang et al. defined a new information entropy to measure the uncertainty of the incomplete information systems [14] and applied the corresponding conditional entropy to reduce redundant features [13]. Meng et al. [22] developed a heuristic attribute reduction algorithm based on tolerance relation rough set, which was beneficial to attribute reduction in large-scale decision systems.

To further increase the efficiency of reduction algorithms, many researchers studied acceleration mechanisms. Because partition (or equivalence) relation is one of the important and primitive notions in the reduction approaches, the computation of equivalence class is one of the key steps for attribute reduction. Hence, Liu et al. [18] proposed the approach of computing the equivalence classes by quick sort, and designed a reduction algorithm based on the positive region with time complexity $O(|C|^2|U|\log|U|)$, where C represents the conditional attribute set, U denotes the object set, and $|\cdot|$ denotes the cardinality of set. Xu et al. [36] computed equivalence classes by radix sort, which improved the performance of attribute reduction algorithms and made heuristic reduction algorithm's time complexity to reduce $\max(O(|C||U|), O(|C|^2|U/C|))$. Qian et al. [26] also presented a counting sort algorithm to deal with inconsistent decision tables and compute positive regions and core attributes, and a hybrid attribute reduction algorithm based on the relationship between indiscernibility and discernibility was proposed with time complexity no more than $\max(O(|C||U|), O(|C|^2|U/C|))$. To further reduce computational time, by using four types of common heuristic reduction algorithms, Qian et al. [27,28] studied an accelerator strategy for the positive-region reduct and three types of entropy reducts in the complete and incomplete decision tables. The heuristic reduction methods based on the accelerator could significantly decrease the time consumed and obtain the same reduct as their original methods. On the basis of the above research, Liang et al. [12] developed a new accelerator that simultaneously decreased the size of universe and reduced the number of attributes in each iteration process of attribute reduction to further accelerate attribute reduction efficiency. In addition, Li et al. [11] proposed judgment theorems for the assignment reduct, distribution reduct, and maximum distribution reduct, derived three new types of attribute significance measure based on judgment theorems, and constructed three quick algorithms corresponding to the three types of reduct.

On the basis of the aforementioned analysis, although several different types of attribute reducts have been studied, these algorithms mainly focus on studying the methods for a single type or only a handful of types of attribute reducts, and there lacks a common reduction framework to obtain various types of attribute reducts for different applicants or user requirements. In addition, some reduction approaches concentrate on seeking quick reduction algorithms to obtain a reduct for large data sets; moreover, these methods are still inefficient and unsuitable for the voluminous and high-dimensional data sets, such as gene data set and image data set. Therefore, it is urgent to develop an approach that integrates the aforementioned two aspects to construct a general and efficient reduction algorithm to meet the requirements of various types of attribute reducts for different criteria.

Out of this motivation, in this study, we develop quick general reduction algorithms for inconsistent decision tables from the viewpoint of the relative discernibility. Two major contributions are included in this study: (1) to combine the definitions of five representative reducts into a unified framework from the viewpoint of the relative discernibility and (2) to study the quick reduction algorithms from the viewpoint of the relative discernibility for the general decision table by a new acceleration strategy. First, a general definition of the decision table for obtaining five representative reducts (relative relation reduct, positive-region reduct, distribution reduct, maximum distribution reduct, and assignment reduct) is presented, and a general reduction approach based on the general discernibility matrix is provided; however, its efficiency is quite unbearable for large data sets. Therefore, on the basis of the definition of the general decision table, the notions and properties of the relative discernibility are proposed and core attributes and attribute significance are studied from the viewpoint of the relative discernibility. Subsequently, two general reduction algorithms based on the relative discernibility, that is, forward reduction strategy and backward reduction strategy, are introduced. To further increase the efficiency of reduction algorithms, quick general reduction algorithms based on the relative discernibility for five types of reducts are designed mainly by reducing the radix sort times in the process of the equivalence class partitioning. Finally, a series of comparative experiments with UCI data sets are conducted to illustrate the efficiency and effectiveness of the proposed reduction algorithms.

The remainder of this study is organized as follows. Section 2 reviews preliminary notions related to the reduct and presents a general model of the decision table and a general reduction approach based on discernibility matrix. Section 3 introduces the corresponding concepts of the relative discernibility and the computational approach for the relative discernibility, and gives the notion of the relative discernibility reduct as well as corresponding propositions. Section 4 presents two general heuristic reduction algorithms based on the relative discernibility for five types of classic reducts. Section 5 investigates the acceleration strategy for attribute reduction and develops two quick general heuristic reduction algorithms. Section 6 constructs a series of comparative experiments to evaluate the performance of our proposed reduction algorithms. Finally, Section 7 presents the conclusions drawn from this study.

2. Preliminaries

In this section, we review several basis concepts in the rough set theory and present a common frame of the decision table for different reduct notions and a general reduction method based on the general discernibility matrix.

Let $\mathrm{DT} = (U, A = C \cup D, V, f)$ be a decision table system, where $U = \{x_1, x_2, \ldots, x_n\}$ is a nonempty finite set of objects called the universe; $A = \{a_1, a_2, \ldots, a_m\}$ is a nonempty and finite set of attributes; $V = \{V_a | \forall a \in A\}$ is a set of value domain of attributes, where V_a is a value set of the attribute a; and $f : U \times A \to V$ is called an information function ($f(x, a) \in V_a$ for each $a \in A$). $C \cap D = \Phi$, where C and D are called the conditional attribute and decision attribute sets, respectively.

For any $R \subseteq A$, the indiscernibility relation is defined as

$$\mathrm{IND}(R) = \left\{(x, y) \in U \times U | f(x, a) = f(y, a),\ \forall a \in R\right\},$$

where x and y are indiscernible with respect to R, only if they have the same values on all attributes in R. The relation $\mathrm{IND}(R)$ is reflexive, symmetric, and transitive, and hence is an equivalence relation. $U/\mathrm{IND}(R) = \{[x]_R | x \in U\}$ (just as U/R) indicates the partitions of U induced by R, which denotes the equivalence class determined by x with respect to R, that is, $[x]_R = \{y | y \in U,\ (x, y) \in \mathrm{IND}(R)\}$.

For any $R \subseteq A$, the relative indiscernibility relation defined by the conditional attribute subset R with respect to the decision attribute D is defined as

$$\mathrm{IND}(R|D) = \left\{(x, y) \in U \times U | \forall a \in R,\ f(x, a) = f(y, a) \vee f(x, D) = f(y, D)\right\}.$$

The relative discernibility relation defined by R with respect to the decision attribute D is defined as

$$\mathrm{DIS}(R|D) = \left\{(x, y) \in U \times U | \exists a \in R,\ f(x, a) \neq f(y, a) \wedge f(x, D) \neq f(y, D)\right\}.$$

Because the relative indiscernibility relation is not satisfied with transitive function, the relative indiscernibility relation is not an equivalence relation. However, the relative discernibility relation is an equivalence relation.

Given a decision table $\mathrm{DT} = (U, C \cup D, V, f)$, where DT is an inconsistent decision table, if there exist $x, y \in U$ $(x \neq y)$, $f(x, C) = f(y, C)$, and $f(x, D) \neq f(y, D)$; otherwise, DT is a consistent decision table. We can also say that DT is inconsistent, if there exists an object pair $(x, y) \in \mathrm{IND}(C)$ and $(x, y) \notin \mathrm{IND}(D)$; otherwise, it is consistent.

For any $X \subseteq U$ and $R \subseteq C$, the pair $(\underline{R}X, \overline{R}X)$ is defined as the rough set approximation of X with respect to R, where $\underline{R}X$ and $\overline{R}X$ denote the lower and upper approximations of X with respect to R, respectively. $\mathrm{POS}_R(X) = \underline{R}X$ is called the positive region of X with respect to R and $\mathrm{POS}_R(D) = \bigcup_{X \in U/D} \underline{R}X$ is called the positive region of D with respect to R.

Given a decision table $\mathrm{DT} = (U, C \cup D, V, f)$, Hu's discernibility matrix $\boldsymbol{M} = \{m(x, y)\}$ is a $|U| \times |U|$ matrix, in which the element $m(x, y)$ is satisfied with ref. [6]:

$$m(x, y) = \begin{cases} \{a|a \in C\} & f(x, a) \neq f(y, a) \wedge f(x, D) \neq f(y, D) \\ \Phi & \text{otherwise} \end{cases},$$

where $m(x, y) \neq \Phi$ indicates that objects x and y are to be distinguished by the conditional attribute set C and the object pair (x, y) is a relative discernibility relation of D with respect to C.

Given a decision table $\mathrm{DT} = (U, C \cup D, V, f)$, let $R \subseteq C$ and R be a reduct based on Hu's discernibility matrix $\boldsymbol{M}$. R should satisfy the following two conditions.

(1) $\forall \Phi \neq m(x, y) \in \boldsymbol{M},\ R \cap (x, y) \neq \Phi$;
(2) $\forall a \in R,\ \exists \Phi \neq m(x, y) \in \boldsymbol{M},\ R - \{a\} \cap (x, y) = \Phi$.

Remark. The reduct results of Hu's discernibility matrix is equivalent to the reduct results of the relative discernibility relationship [6,20]. Hence, in this study, the relative relation reduct (R-reduct) is also called as Hu's discernibility matrix reduct (H-reduct).

2.1. Relative attribute reduction

Given a decision table $\mathrm{DT} = (U, C \cup D, V, f)$, let $R \subseteq C$ and $U/D = \{Y_1, Y_2, \ldots, Y_m\}$, where $m = |U/D|$. For any $x \in U$, the membership distribution function of object x on all decision class with respect to $R \subseteq C$ is defined as

$$\mu_R(x) = \big(P\big(Y_1|[x]_R\big), P\big(Y_2|[x]_R\big), \ldots, P\big(Y_m|[x]_R\big)\big),$$

where $Y_j \in U/D$ $(1 \leq j \leq m)$ and $P(Y_j|[x]_R) = \frac{|[x]_R \cap y_j|}{|[x]_R|}$.

Let $\delta_R(x) = \{Y_j | Y_j \cap [x]_R \neq \Phi\}$ $(1 \leq j \leq m)$ be the family of decision equivalence classes that have an intersection with $[x]_R$. Let $\gamma_R(x) = \{Y_j | Y_j = \mathrm{argmax}_{1 \leq i \leq m} P(Y_i|[x]_R)\}$ be the decision equivalence class that has maximum rough member of object x given in $[x]_R$ with respect to all of the decision class.

In inconsistent decision tables, for different user requirements in different applications, we can obtain different types of attribute reducts. Zhou et al. [45] studied the relationships among 12 types of the relative reduct (where absolute reduct is not a relative reduct) for inconsistent decision tables, where there are only five intrinsically different relative reducts, namely relative relation reduct (Hu's discernibility matrix reduct, denoted as H-reduct), positive-region reduct, distribution reduction, maximum distribution reduct, and assignment reduct. In the following, we provide five representative relative reduct definitions:

Definition 1. Let $\mathrm{DT} = (U, C \cup D, V, f)$ be a decision table and $R \subseteq C$. Then, we have five classical relative reducts as follows:

(1) R is a relative relation reduct (denoted as R-reduct) of DT, only if $\mathrm{IND}(R|D) = \mathrm{IND}(C|D)$ and $\forall R' \subset R,\ \mathrm{IND}(R'|D) \neq \mathrm{IND}(C|D)$;
(2) R is a positive-region reduct (denoted as P-reduct) of DT, only if $\mathrm{POS}_R(D) = \mathrm{POS}_C(D)$ and $\forall R' \subset R,\ \mathrm{POS}_{R'}(D) \neq \mathrm{POS}_C(D)$;
(3) R is a distribution reduct (denoted as D-reduct) of DT, only if $\forall x \in U,\ \mu_R(x) = \mu_C(x)$, and $\forall R' \subset R,\ \exists y \in U,\ \mu_{R'}(y) \neq \mu_C(y)$;
(4) R is a maximum distribution reduct (denoted as Md-reduct) of DT, only if $\forall x \in U,\ \gamma_R(x) = \gamma_C(x)$, and $\forall R' \subset R,\ \exists y \in U,\ \gamma_{R'}(y) \neq \gamma_C(y)$;
(5) R is an assignment reduct (denoted as A-reduct) of DT, only if $\forall x \in U,\ \delta_R(x) = \delta_C(x)$, and $\forall R' \subset R,\ \exists y \in U,\ \delta_{R'}(y) \neq \delta_C(y)$.

R-reduct is the minimal set of conditional attributes that preserves the invariance of the relative indiscernibility relation of conditional attribute subset on U. P-reduct is the minimal set of conditional attributes that preserves the same positive region as conditional attribute C on U. D-reduct is the minimum subset of conditional attributes that preserves values of decision function for all objects on U. Md-reduct is the minimum subset of conditional attributes that preserves maximum decision function for all objects on U. A-reduct is the minimum subset of conditional attributes that preserves the invariance of possible decision classes for an arbitrary object on U.

2.2. A general definition of the decision table

On the basis of the above analysis, the definitions of different reducts have been proposed for preserving different properties or meeting different user requirements in the real applications. However, there is a lack of a general definition of attribute reducts. Hence, for five representative reducts, namely R-reduct (or H-reduct), P-reduct, D-reduct, Md-reduct, and A-reduct, five transmutative definitions of the decision table for five reducts are presented, which are called as DT_R (or DT_H), DT_P, DT_D, DT_{Md}, and DT_A, respectively. In this subsection, we first combine five definitions of the decision table into a general framework of the decision table for facilitating uniformly the definitions of various reducts, which is named as $DT_\triangle$, where $\triangle \in \{H = R, P, D, Md, A\}$.

Definition 2. Given a decision table $DT = (U, C \cup D, V, f)$, let the relative relation decision table of DT for H-reduct be $DT_H = (U, C \cup D, V_H, f_H)$, which satisfies

(1) $f_H(x_i, C) = f(x_i, C)$, $\forall x_i \in U$;
(2) $f_H(x_i, D) = f(x_i, D)$, $\forall x_i \in U$.

Definition 3. Given a decision table $DT = (U, C \cup D, V, f)$, let the positive region decision table of DT for P-reduct be $DT_P = (U, C \cup D, V_P, f_P)$, which satisfies

(1) $f_P(x_i, C) = f(x_i, C)$, $\forall x_i \in U$
(2) $f_P(x_i, D) = \begin{cases} MAX_D + 1 & \exists x_j, f(x_i, C) = f(x_j, C) \wedge f(x_i, D) \neq f(x_j, D) \\ f(x_i, D) & \text{otherwise} \end{cases}$,

where $MAX_D = \max\{(f(x, D))\}$.

Definition 4. Given a decision table $DT = (U, C \cup D, V, f)$, let the distribution decision table of DT for D-reduct be $DT_D = (U, C \cup D, V_D, f_D)$, which satisfies

(1) $f_D(x_i, C) = f(x_i, C)$, $\forall x_i \in U$;
(2) $f_D(x_i, D) \in [1 \ldots |U/C|]$, where if $\exists x_i, x_j \in U$, $\delta_C(x_i) \neq \delta_C(x_j)$, then $f_D(x_i, D) \neq f_D(x_j, D)$; otherwise, $f_D(x_i, D) = f_D(x_j, D)$.

Definition 5. Given a decision table $DT = (U, C \cup D, V, f)$, let the max-distribution decision table of DT for Md-reduct be $DT_{Md} = (U, C \cup D, V_{Md}, f_{Md})$, which satisfies

(1) $f_{Md}(x_i, C) = f(x_i, C)$, $\forall x_i \in U$;
(2) $f_{Md}(x_i, D) \in [1...|U/C|]$, where if $\exists x_i, x_j \in U$, $\gamma_C(x_i) \neq \gamma_C(x_j)$, then $f_{Md}(x_i, D) \neq f_{Md}(x_j, D)$; otherwise, $f_{Md}(x_i, D) = f_{Md}(x_j, D)$.

Definition 6. Given a decision table $DT = (U, C \cup D, V, f)$, let the assignment decision table of DT for A-reduct be $DT_A = (U, C \cup D, V_A, f_A)$, which satisfies

(1) $f_A(x_i, C) = f(x_i, C)$, $\forall x_i \in U$;
(2) $f_A(x_i, D) \in [1...|U/D|]$, where if $\exists x_i, x_j \in U$, $\mu_C(x_i) \neq \mu_C(x_j)$, then $f_A(x_i, D) \neq f_A(x_j, D)$; otherwise, $f_A(x_i, D) = f_A(x_j, D)$.

According to the corresponding definitions of the decision table for deriving various reducts, we provide a general decision table definition described as follows, for summarizing the common structure of the different decision tables.

Definition 7. Given a decision table $DT = (U, C \cup D, V, f)$, let the general decision table be $DT_\triangle = (U, C \cup D, V_\triangle, f_\triangle)$ of DT, which satisfies

(1) $f_\triangle(x_i, C) = f(x_i, C)$, $\forall x_i \in U$;
(2) $f_\triangle(x_i, D) \in [1 \ldots k]$, where if $\exists x_i, x_j \in U$, $\triangle(x_i, C) \neq \triangle(x_j, C)$, then $f_\triangle(x_i, D) \neq f_\triangle(x_j, D)$; otherwise, $f_\triangle(x_i, D) = f_\triangle(x_j, D)$,

where the values of k are MAX_D, $MAX_D + 1$, $|U/C|$, $|U/C|$, and $|U/D|$ for $\triangle$ = H, P, D, Md, and A, respectively.

The goal of the general decision table is to facilitate the construction of the transmutative decision table for obtaining $\triangle$-reduct to meet a certain property $\triangle$. Meanwhile, the general decision table can conveniently be unified into attribute reduction algorithms from viewpoints of the discernibility matrix and distinguished ability, and be beneficial to improve implementation of algorithms for different reducts.

Table 1
Inconsistent decision table DT.

U	a	B	c	d	D
x_1	0	1	1	1	1
x_2	1	0	0	1	1
x_3	1	0	0	1	1
x_4	1	0	0	1	2
x_5	1	0	1	1	1
x_6	1	0	1	1	2
x_7	1	0	1	1	2

Table 2
Decision tables for five types of reducts with respect to DT.

U	a	b	c	d	$\mu_C(x)$	$\delta_C(x)$	$\gamma_C(x)$	D_{H}	D_{P}	D_{D}	D_{Md}	D_{A}
x_1	0	1	1	1	$(1,0)$	$\{D_1\}$	D_1	1	1	1	1	1
x_2	1	0	0	1	$(2/3,1/3)$	$\{D_1,D_2\}$	D_1	1	3	2	1	2
x_3	1	0	0	1	$(2/3,1/3)$	$\{D_1,D_2\}$	D_1	1	3	2	1	2
x_4	1	0	0	1	$(2/3,1/3)$	$\{D_1,D_2\}$	D_1	2	3	2	1	2
x_5	1	0	1	1	$(1/3,2/3)$	$\{D_1,D_2\}$	D_2	1	3	3	2	2
x_6	1	0	1	1	$(1/3,2/3)$	$\{D_1,D_2\}$	D_2	2	3	3	2	2
x_7	1	0	1	1	$(1/3,2/3)$	$\{D_1,D_2\}$	D_2	2	3	3	2	2

Example 1. Table 1 is a decision table, where $U=\{x_1,x_2,x_3,x_4,x_5,x_6,x_7\}$, the conditional attribute set $C=\{a,b,c\}$ and the decision attribute D.

From Table 1, $U/C=\{\{x_1\},\{x_2,x_3,x_4\},\{x_5,x_6,x_7\}\}$ and $U/D=\{\{x_1,x_2,x_3,x_5\},\{x_4,x_6,x_7\}\}$, we can get $\mu_C(x_1)=(1,0)$, $\mu_C(x_2)=\mu_C(x_3)=\mu_C(x_4)=(2/3,1/3)$, $\mu_C(x_5)=\mu_C(x_6)=\mu_C(x_7)=(1/3,2/3)$;

$$\delta_C(x_1)=\{D_1\},\qquad \delta_C(x_2)=\delta_C(x_3)=\delta_C(x_4)=\delta_C(x_5)=\delta_C(x_6)=\delta_C(x_7)=\{D_1,D_2\};$$

$$\gamma_C(x_1)=\gamma_C(x_2)=\gamma_C(x_3)=\gamma_C(x_4)=D_1,\qquad \gamma_C(x_5)=\gamma_C(x_6)=\gamma_C(x_7)=D_2.$$

The new decision values of $f_{\mathrm{H}}(x_i,D)$, $f_{\mathrm{P}}(x_i,D)$, $f_{\mathrm{D}}(x_i,D)$, $f_{\mathrm{Md}}(x_i,D)$, and $f_{\mathrm{A}}(x_i,D)$ for every object $x_i\in U$ are computed and shown in Table 2. The new decision table for a certain reduct can be constructed through a combination of the conditional attribute set and the new decision attribute. Hence, we can create five new decision tables, $\mathrm{DT_H}$, $\mathrm{DT_P}$, $\mathrm{DT_D}$, $\mathrm{DT_{Md}}$, and $\mathrm{DT_A}$, which are listed in Table 2.

In Table 2, the combination of the object set $U=\{x_1,x_2,x_3,x_4,x_5,x_6,x_7\}$, the conditional attribute set $C=\{a,b,c,d\}$, and the decision attribute D_{H} can get the decision table $\mathrm{DT_H}$; the combination of the object set U, the conditional attribute set C, and the decision attribute D_{P} can obtain the decision table $\mathrm{DT_P}$; the combination of the object set U, the conditional attribute set C, and the decision attribute D_{D} can get the decision table $\mathrm{DT_D}$; the combination of the object set U, the conditional attribute set C, and the decision attribute D_{Md} can obtain the decision table $\mathrm{DT_{Md}}$; and the combination of the object set U, the conditional attribute set C, and the decision attribute D_{A} can get the decision table $\mathrm{DT_A}$.

2.3. General definition of the discernibility matrix and discernibility function

A reduct is a minimal subset of the entire conditional attribute set that are jointly sufficient and individually necessary for preserving a certain property of the original decision table [20,43,45]. On the basis of the general decision table $\mathrm{DT}_\triangle$, the general definition of reduct can be described as follows:

Definition 8. Given the general decision table $\mathrm{DT}_\triangle=(U,C\cup D,V_\triangle,f_\triangle)$ and a certain property $\triangle$ of the decision table, the attribute subset $R\subseteq C$ is a $\triangle$-reduct of C with respect to D, if it satisfies the following two conditions:

(1) $\triangle(R)=\triangle(C)$;
(2) $\forall R'\subset R$, $\triangle(R')\neq\triangle(C)$.

The first condition ensures that a certain property is preserved by a subset R of conditional attributes. The second condition ensures that the constructed attribute set R is the minimum. The purpose of the general reduct is to find the minimal attribute subset of conditional attributes that can keep a certain property or criterion unchanged.

Similar to Hu's discernibility matrix [6], we can define a general discernibility matrix for the general decision table $\mathrm{DT}_\triangle=(U,C\cup D,V_\triangle,f_\triangle)$.

Definition 9. Given a general decision table $\mathrm{DT}_\triangle = (U, C \cup D, V_\triangle, f_\triangle)$ and a certain property $\triangle$ of the decision table, the general $\triangle$-discernibility matrix $\boldsymbol{M}_\triangle = \{m_\triangle(x_i, x_j)\}$ of the property $\triangle$ is a $|U| \times |U|$ matrix, where $m_\triangle(x_i, x_j)$ satisfies

$$m_\triangle(x_i, x_j) = \begin{cases} \{a | a \in C\} & f_\triangle(x_i, a) \neq f_\triangle(x_j, a) \wedge f_\triangle(x_i, D) \neq f_\triangle(x_j, D) \\ \Phi & \text{otherwise} \end{cases}$$

The matrix element $m_\triangle(x_i, x_j) \neq \Phi$ indicates that object pair (x_i, x_j) can be distinguished by any attribute in $m_\triangle(x_i, x_j)$.

Theorem 1. *Given the decision table DT and the general decision table* $\mathrm{DT}_\triangle$ *of DT,* $R \subseteq C$ *is Hu's discernibility matrix reduct of the general decision table* $\mathrm{DT}_\triangle$*, and then* $R \subseteq C$ *is a* $\triangle$*-reduct of the original decision table DT.*

Proof. (1) Let $\triangle = \mathrm{H}$. H-reduct decision table $\mathrm{DT_H}$ is the same as the decision table DT. Therefore, H-reduct of $\mathrm{DT_H}$ is H-reduct of the original decision table DT.

(2) Let $\triangle = \triangle'$ ($\triangle' =$ P, D, Md, or A) and $R \subseteq C$ be Hu's discernibility matrix reduct of the general decision table $\mathrm{DT}_\triangle$.

First, prove that $\forall \Phi \neq m_{\triangle'}(x_i, x_j) \in \boldsymbol{M}_{\triangle'}$, $R \cap m_{\triangle'}(x_i, x_j) \neq \Phi \Rightarrow \triangle'(R) = \triangle'(C)$.

Proof by contradiction. Suppose that $\triangle'(R) \neq \triangle'(\mathrm{C})$. In other words, there exists $x_i \in U$ which satisfies $\triangle'(C)$, but does not satisfy with $\triangle'(R)$. That is, there exist $x_j \in U$ $(i \neq j)$, such that $f_{\triangle'}(x_i, C) \neq f_{\triangle'}(x_j, C) \wedge f_{\triangle'}(x_i, D) \neq f_{\triangle'}(x_j, D)$, but $f_{\triangle'}(x_i, R) = f_{\triangle'}(x_j, R)$. Hence, $C \cap m_{\triangle'}(x_i, x_j) \neq \Phi$, but $R \cap m_{\triangle'}(x_i, x_j) = \Phi$. There must exist $a \in C - R$, which satisfies $f_{\triangle'}(x_i, a) \neq f_{\triangle'}(x_j, a) \wedge f_{\triangle'}(x_i, D) \neq f_{\triangle'}(x_j, D)$, that is, $a \in m_{\triangle'}(x_i, x_j)$. Hence, we have $\Phi \neq m_{\triangle'}(x_i, x_j) \in \boldsymbol{M}_{\triangle'}$ and $R \cap m_{\triangle'}(x_i, x_j) = \Phi$, which is contradictory with the premise. Therefore, $\Rightarrow$ holds.

Second, prove that for every $a \in R$, there exists $\Phi \neq m_{\triangle'}(x_i, x_j) \in \boldsymbol{M}_{\triangle'}$, which satisfies $m_{\triangle'}(x_i, x_j) \cap R - \{a\} = \Phi \Rightarrow \triangle'(C) \neq \triangle'(R - \{a\})$.

Proof by contradiction. Suppose that there exists $a \in R$, which satisfies $\triangle'(C) = \triangle'(R - \{a\})$. From the assumed conditions, $x_i \in U$ satisfies $\triangle'(C)$ and also with $\triangle'(R - \{a\})$, such that for every $x_i, x_j \in U$ $(i \neq j)$, if $f_{\triangle'}(x_i, C) \neq f_{\triangle'}(x_j, C) \wedge f_{\triangle'}(x_i, D) \neq f_{\triangle'}(x_j, D)$, then $f_{\triangle'}(x_i, R - \{a\}) \neq f_{\triangle'}(x_j, R - \{a\}) \wedge f_{\triangle'}(x_i, D) \neq f_{\triangle'}(x_j, D)$. In addition, for every $\Phi \neq m_{\triangle'}(x_i, x_j) \in \boldsymbol{M}_{\triangle'}$, we have $C \cap m_{\triangle'}(x_i, x_j) \neq \Phi$, that is, $f_{\triangle'}(x_i, C) \neq f_{\triangle'}(x_j, C) \wedge f_{\triangle'}(x_i, D) \neq f_{\triangle'}(x_j, D)$. Hence, $R - \{a\} \cap m_{\triangle'}(x_i, x_j) \neq \Phi$, which is in conflict with the premise. Therefore, $\Rightarrow$ holds. □

Definition 10. Given the general decision table $\mathrm{DT}_\triangle = (U, C \cup D, V_\triangle, f_\triangle)$ and the general $\triangle$-discernibility matrix $\boldsymbol{M}_\triangle = \{m_\triangle(x_i, x_j)\}$, if $|m_\triangle(x_i, x_j)| = 1$ and $a \in m_\triangle(x_i, x_j)$, then the attribute a is a core attribute of $\mathrm{DT}_\triangle$ with respect to $\triangle$-discernibility matrix, described as $a \in \mathrm{MCORE}_\triangle(C)$.

Definition 11. The discernibility function of general $\triangle$-discernibility matrix $\boldsymbol{M}_\triangle$ is defined as

$$DF(\boldsymbol{M}_\triangle) = \wedge\left\{\vee m_\triangle(x_i, x_j) | \forall x_i, x_j \in U,\ m_\triangle(x_i, x_j) \neq \Phi\right\},$$

where $\vee m_\triangle(x_i, x_j)$ represents the logical disjunction of all attributes in $\boldsymbol{M}_\triangle$, which indicates that the object pair (x_i, x_j) can be discernible regarding any attribute in $m_\triangle(x_i, x_j)$, $\wedge\{\vee m_\triangle(x_i, x_j)\}$ represents the conjunction of all $\vee m_\triangle(x_i, x_j)$, which indicates that the family of discernible object pairs can be discernible regarding a set of attributes.

The discernibility function can be used to state an important result with regard to the reduct set of an information table [31].

Example 2 *(Continued from Example 1).* We can construct $\boldsymbol{M}_\mathrm{H}$, $\boldsymbol{M}_\mathrm{P}$, $\boldsymbol{M}_\mathrm{D}$, $\boldsymbol{M}_\mathrm{Md}$, and $\boldsymbol{M}_\mathrm{A}$, and obtain $\mathrm{MCORE_H}(C) = \{c\}$, $\mathrm{MCORE_P}(C) = \Phi$, $\mathrm{MCORE_D}(C) = \{c\}$, $\mathrm{MCORE_{Md}}(C) = \{c\}$, and $\mathrm{MCORE_A}(C) = \Phi$. Thus, we have

$DF(\boldsymbol{M}_\mathrm{H}) = (a \vee b \vee c) \wedge (a \vee b) \wedge (c)$, and both of $\{ac\}$ and $\{bc\}$ are H-reduct (or R-reduct) of DT;
$DF(\boldsymbol{M}_\mathrm{P}) = (a \vee b \vee c) \wedge (a \vee b)$, and both of $\{a\}$ of and $\{b\}$ are P-reduct of DT;
$DF(\boldsymbol{M}_\mathrm{D}) = (a \vee b \vee c) \wedge (a \vee b) \wedge (c)$, and both of $\{ac\}$ and $\{bc\}$ are D-reduct of DT;
$DF(\boldsymbol{M}_\mathrm{Md}) = (a \vee b \vee c) \wedge (a \vee b) \wedge (c)$, and both of $\{ac\}$ and $\{bc\}$ are Md-reduct of DT;
$DF(\boldsymbol{M}_\mathrm{A}) = (a \vee b \vee c) \wedge (a \vee b)$, and both of $\{a\}$ and $\{b\}$ are A-reduct of DT.

3. Relative discernibility and the corresponding reduct

The discernibility matrix can directly describe the relation of object pairs. The reduct of the discernibility matrix is a minimum attribute set, which retains the discernibility relation defined by the entire attribute set. Unfortunately, the discernibility matrix-based reduction method is commonly computationally expensive, and it is quite intolerable for dealing with large-scale and high-dimensional data sets. Therefore, in this section, the relative discernibility and the corresponding properties are introduced first. Next, the reduct definition based on the relative discernibility with respect to the general decision table $\mathrm{DT}_\triangle$ is presented. It is proved that the reduct of the relative discernibility is identical to the reducts of the corresponding discernibility matrix. Then, the concepts and properties of attribute significance and core attributes based on the relative discernibility are studied for the general decision table.

3.1. General relative discernibility

Definition 12. Let $\mathrm{DT}_\triangle = (U, A = C \cup D, V_\triangle, f_\triangle)$ be the general decision table, $R \subseteq A$, the discernibility and indiscernibility of the attribute set R are defined by

$$W_\triangle(R) = \left|\mathrm{DIS}_\triangle(R)\right| = \left|\left\{(x_i, x_j) \in U \times U \mid \exists a \in R,\ f_\triangle(x_i, a) \neq f_\triangle(x_j, a)\right\}\right|$$
$$\overline{W_\triangle(R)} = \left|\mathrm{IND}_\triangle(R)\right| = \left|\left\{(x_i, x_j) \in U \times U \mid \forall a \in R,\ f_\triangle(x_i, a) = f_\triangle(x_j, a)\right\}\right|,$$

where $W_\triangle(R)$ denotes the discernibility of the attribute set R and is the measure of the distinguishing ability of attribute set R and $\overline{W_\triangle(R)}$ denotes the indiscernibility of the attribute set R and is the measure of the indistinguishing ability of attribute set R.

Definition 13. Let $\mathrm{DT}_\triangle = (U, C \cup D, V_\triangle, f_\triangle)$ be the general decision table, $a \in C$ and $R \subseteq C$, the relative relations with respect to decision attribute D are defined by

$$\mathrm{DIS}_\triangle(a|D) = \left\{(x_i, x_j) \in U \times U \mid f_\triangle(x_i, a) \neq f_\triangle(x_j, a) \wedge f_\triangle(x_i, D) \neq f_\triangle(x_j, D)\right\}$$
$$\mathrm{DIS}_\triangle(R|D) = \left\{(x_i, x_j) \in U \times U \mid \exists a \in R,\ f_\triangle(x_i, a) \neq f_\triangle(x_j, a) \wedge f_\triangle(x_i, D) \neq f_\triangle(x_j, D)\right\}$$
$$\mathrm{IND}_\triangle(a|D) = \left\{(x_i, x_j) \in U \times U \mid f_\triangle(x_i, a) = f_\triangle(x_j, a) \vee f_\triangle(x_i, D) = f_\triangle(x_j, D)\right\}$$
$$\mathrm{IND}_\triangle(R|D) = \left\{(x_i, x_j) \in U \times U \mid \forall a \in R,\ f_\triangle(x_i, a) = f_\triangle(x_j, a) \vee f_\triangle(x_i, D) = f_\triangle(x_j, D)\right\},$$

where $\mathrm{DIS}_\triangle(a|D)$ denotes the relative discernibility relation of the decision attribute D with respect to the conditional attribute a; $\mathrm{DIS}_\triangle(R|D)$ denotes the relative discernibility relation of the decision attribute D with respect to the conditional attribute set R; $\mathrm{IND}_\triangle(a|D)$ represents the relative indiscernibility relation of the decision attribute D with respect to the conditional attribute a; and $\mathrm{IND}_\triangle(R|D)$ denotes the relative indiscernibility relation of the decision attribute D with respect to the conditional attribute set R.

Definition 14. Let $\mathrm{DT}_\triangle = (U, C \cup D, V_\triangle, f_\triangle)$ be the general decision table, $a \in C$ and $R \subseteq C$, the relative discernibility and indiscernibility on the decision attribute D are defined by

$$W_\triangle(a|D) = \left|\mathrm{DIS}_\triangle(a|D)\right| = \left|\left\{(x_i, x_j) \in U \times U \mid f_\triangle(x_i, a) \neq_\triangle f(x_j, a) \wedge f_\triangle(x_i, D) \neq f_\triangle(x_j, D)\right\}\right|$$
$$W_\triangle(R|D) = \left|\mathrm{DIS}_\triangle(R|D)\right| = \left|\left\{(x_i, x_j) \in U \times U \mid \exists a \in R,\ f_\triangle(x_i, a) \neq f_\triangle(x_j, a) \wedge f_\triangle(x_i, D) \neq f_\triangle(x_j, D)\right\}\right|$$
$$\overline{W_\triangle(a|D)} = \left|\mathrm{IND}_\triangle(a|D)\right| = \left|\left\{(x_i, x_j) \in U \times U \mid f_\triangle(x_i, a) = f_\triangle(x_j, a) \vee f_\triangle(x_i, D) = f_\triangle(x_j, D)\right\}\right|$$
$$\overline{W_\triangle(R|D)} = \left|\mathrm{IND}_\triangle(R|D)\right| = \left|\left\{(x_i, x_j) \in U \times U \mid \forall a \in R,\ f_\triangle(x_i, a) = f_\triangle(x_j, a) \vee f_\triangle(x_i, D) = f_\triangle(x_j, D)\right\}\right|,$$

where $W_\triangle(a|D)$ denotes the relative discernibility of the decision attribute D with respect to the conditional attribute a; $W_\triangle(R|D)$ represents the relative discernibility of the decision attribute D with respect to the conditional attribute set R; $\overline{W_\triangle(a|D)}$ represents the relative indiscernibility of the decision attribute D with respect to the conditional attribute a; and $\overline{W_\triangle(R|D)}$ denotes the relative indiscernibility of the decision attribute D with respect to the conditional attribute set R.

Proposition 1. *Let $\mathrm{DT}_\triangle = (U, C \cup D, V_\triangle, f_\triangle)$ be the general decision table, and $R \subseteq C$. We have*

(1) $\mathrm{DIS}_\triangle(R|D) \cap \mathrm{IND}_\triangle(R|D) = \Phi$;
(2) $\mathrm{DIS}_\triangle(R|D) \cup \mathrm{IND}_\triangle(R|D) = U \times U$.

Proposition 2. *If $\mathrm{DT}_\triangle = (U, C \cup D, V_\triangle, f_\triangle)$ is the general decision table, let $P \subseteq R \subseteq C$, then we have*

(1) $W_\triangle(R|D) + \overline{W_\triangle(R|D)} = |U|^2$;
(2) $W_\triangle(P|D) \leq W_\triangle(R|D)$;
(3) $\overline{W_\triangle(P|D)} \geq \overline{W_\triangle(R|D)}$.

Fig. 1 shows the relationship between objects pair of the universe, discernibility relation, indiscernibility relation, and relative indiscernibility relation.

Proposition 3. *Let $\mathrm{DT}_\triangle = (U, C \cup D, V_\triangle, f_\triangle)$ be the general decision table and $R \subseteq C$, we have*

$$W_\triangle(R|D) = |U|^2 - \overline{W_\triangle(D)} - \overline{W_\triangle(R)} + \overline{W_\triangle(D \cup R)}.$$

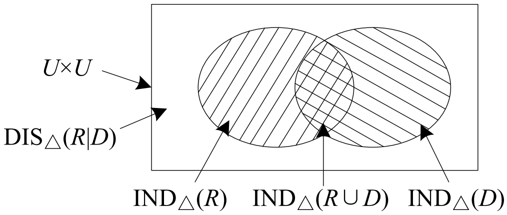

Fig. 1. Discernibility relation, indiscernibility relation, and relative discernibility relation.

Proposition 4. *Let* $\mathrm{DT}_\triangle = (U, C \cup D, V_\triangle, f_\triangle)$ *be the general decision table and* $R \subseteq C$, $U/R = \{X_1, X_2, \ldots, X_n\}$, *and* $U/D = \{Y_1, Y_2, \ldots, Y_m\}$, *we have* $\overline{W_\triangle(R)} = \sum_{i=1}^{n} |X_i|^2$ *and* $\overline{W_\triangle(D)} = \sum_{j=1}^{m} |Y_j|^2$.

Proposition 5. *Let* $\mathrm{DT}_\triangle = (U, C \cup D, V_\triangle, f_\triangle)$ *be the general decision table and* $R \subseteq C$, $U/R = \{X_1, X_2, \ldots, X_n\}$, $U/D = \{Y_1, Y_2, \ldots, Y_m\}$, *and* $U/(D \cup R) = \{Z_1, Z_2, \ldots, Z_t\}$, *we have*

$$W_\triangle(R|D) = |U|^2 - \overline{W_\triangle(D)} - \overline{W_\triangle(R)} + \overline{W_\triangle(D \cup R)} = |U|^2 - \sum_{j=1}^{m} |Y_j|^2 - \sum_{i=1}^{n} |X_i|^2 + \sum_{k=1}^{t} |Z_k|^2.$$

Example 3. For the decision table in Table 2, let $\triangle = \mathrm{H}$, $R_1 = \{a\}$, and $R_2 = \{a, c\}$. $U/D_\mathrm{H} = \{\{x_1, x_2, x_3, x_6\}, \{x_4, x_5, x_7\}\}$, $U/R_1 = \{\{x_1\}, \{x_2, x_3, x_4, x_5, x_6, x_7\}\}$, $U/R_2 = \{\{x_1\}, \{x_2, x_3, x_4\}, \{x_5, x_6, x_7\}\}$, $U/(D_\mathrm{H} \cup R_1) = \{\{x_1\}, \{x_2, x_3, x_5\}, \{x_4, x_6, x_7\}\}$, and $U/(D_\mathrm{H} \cup R_2) = \{\{x_1\}, \{x_2, x_3\}, \{x_4\}, \{x_5\}, \{x_6, x_7\}\}$.

We have the following results:

$$\overline{W_\mathrm{H}(D)} = 4^2 + 3^2 = 25, \qquad \overline{W_\mathrm{H}(R_1)} = 1^2 + 6^2 = 37, \qquad \overline{W_\mathrm{H}(D_\mathrm{H} \cup R_1)} = 1^2 + 2 * 3^2 = 19,$$
$$\overline{W_\mathrm{H}(R_2)} = 19, \quad \text{and} \quad \overline{W_\mathrm{H}(D_\mathrm{H} \cup R_2)} = 11.$$
$$W_\mathrm{H}(R_1|D) = |U|^2 - \overline{W_\mathrm{H}(D)} - \overline{W_\mathrm{H}(R_1)} + \overline{W_\mathrm{H}(D_\mathrm{H} \cup R_1)} = 49 - 25 - 37 + 19 = 6.$$
$$W_\mathrm{H}(R_2|D) = |U|^2 - \overline{W_\mathrm{H}(D)} - \overline{W_\mathrm{H}(R_2)} + \overline{W_\mathrm{H}(D_\mathrm{H} \cup R_2)} = 49 - 25 - 19 + 11 = 16.$$

Thus, $W_\mathrm{H}(R_1|D) < W_\mathrm{H}(R_2|D)$.

3.2. Reduct and core attributes for general relative discernibility

Definition 15. Given the general decision table $\mathrm{DT}_\triangle = (U, C \cup D, V_\triangle, f_\triangle)$. $Red_\triangle$ is the reduct set of $\mathrm{DT}_\triangle$ with respect to relative discernibility. $R \subseteq C$, $R \in Red_\triangle$ satisfies the following two conditions:

(1) $W_\triangle(R|D) = W_\triangle(C|D)$
(2) $\forall a \in R$, $W_\triangle(R - \{a\}|D) < W_\triangle(R|D)$.

The first condition ensures that the relative discernibility can be preserved by the attribute set R. The second condition indicates that each attribute in R is necessary with regard to the relative discernibility. From Definition 15, the attribute reduct of the relative discernibility shows to find a subset of attribute that has the same discernibility as the original data set without redundant attributes.

Theorem 2. *The relative discernibility reduct is equivalent to Hu's discernibility matrix reduct.*

(1) First, prove that if $R \subseteq C$, $\forall \Phi \neq m_\triangle(x_i, x_j) \in \boldsymbol{M}_\triangle$, $R \cap m_\triangle(x_i, x_j) \neq \Phi \Leftrightarrow W_\triangle(R|D) = W_\triangle(C|D)$.

$\Rightarrow$. Proof by contradiction. Suppose that $W_\triangle(R|D) < W_\triangle(C|D)$, from Definition 14 and Proposition 2, we can know that there at least exist $x_i, x_j \in U$ $(i \neq j)$, such that $f_\triangle(x_i, R) = f_\triangle(x_j, R) \wedge f_\triangle(x_i, D) \neq f_\triangle(x_j, D)$, but $f_\triangle(x_i, C) \neq f_\triangle(x_j, C)$. Hence, there must exist $a \in C - R$, which satisfies $f_\triangle(x_i, a) \neq f_\triangle(x_j, a) \wedge f_\triangle(x_i, D) \neq f_\triangle(x_j, D)$, that is, $a \in m_\triangle(x_i, x_j)$. Hence, we have $\Phi \neq m_\triangle(x_i, x_j) \in \boldsymbol{M}_\triangle$ and $R \cap m_\triangle(x_i, x_j) = \Phi$, which is contradictory with antecedent of (1). Therefore $\Rightarrow$ holds.

$\Leftarrow$. Proof by contradiction. Assume that there exists $\Phi \neq m_\triangle(x_i, x_j) \in \boldsymbol{M}_\triangle$ and $R \cap m_\triangle(x_i, x_j) = \Phi$, then we have $f_\triangle(x_i, R) = f_\triangle(x_j, R) \wedge f_\triangle(x_i, D) \neq f_\triangle(x_j, D)$. From Definition 12, we can obtain $(x_i, x_j) \in \mathrm{IND}_\triangle(R|D)$, that is, $(x_i, x_j) \notin \mathrm{DIS}_\triangle(R|D)$. Because of $\Phi \neq m_\triangle(x_i, x_j) \in \boldsymbol{M}_\triangle$, there must exist $a \in C - R$, which satisfies $f_\triangle(x_i, a) \neq f_\triangle(x_j, a) \wedge f_\triangle(x_i, D) \neq f_\triangle(x_j, D)$, that is, $f_\triangle(x_i, C) \neq f_\triangle(x_j, C)$, such that $W_\triangle(R|D) \neq W_\triangle(C|D)$, which is contradictory with consequent of (1). Hence, $\Leftarrow$ holds.

(2) Second, if $R \subseteq C$, for every $a \in R$, there exists $\Phi \neq m_\triangle(x_i, x_j) \in \boldsymbol{M}_\triangle$, which satisfies $m_\triangle(x_i, x_j) \cap R - \{a\} = \Phi \Leftrightarrow W_\triangle(R - \{a\}|D) < W_\triangle(R|D)$.

$\Rightarrow$. Proof by contradiction. Suppose that there exists $a \in R$, which satisfies $W_\triangle(R - \{a\}|D) = W_\triangle(R|D)$. From (1), we know that for every $\Phi \neq m_\triangle(x_i, x_j) \in \boldsymbol{M}_\triangle$, we have $R \cap m_\triangle(x_i, x_j) \neq \Phi$, that is, $f_\triangle(x_i, R) \neq f_\triangle(x_j, R) \wedge f_\triangle(x_i, D) \neq f_\triangle(x_j, D)$. Because of $W_\triangle(R-\{a\}|D) = W_\triangle(R|D)$, from Definition 14, we know $f_\triangle(x_i, R-\{a\}) = f_\triangle(x_j, R-\{a\}) \wedge f_\triangle(x_i, D) \neq f_\triangle(x_j, D)$, such that $m_\triangle(x_i, x_j) \cap R - \{a\} \neq \Phi$, which is contradictory with antecedent of (2). Therefore, $\Rightarrow$ holds.

$\Leftarrow$. Proof by contradiction. Suppose that there exists $a \in R$, for every $\Phi \neq m_\triangle(x_i, x_j) \in \boldsymbol{M}_\triangle$, we have $m_\triangle(x_i, x_j) \cap R - \{a\} \neq \Phi$, then we know $f_\triangle(x_i, R - \{a\}) \neq f_\triangle(x_j, R - \{a\}) \wedge f_\triangle(x_i, D) \neq f_\triangle(x_j, D)$. From Definition 12, we can get $(x_i, x_j) \notin \mathrm{DIS}_\triangle(R - \{a\}|D)$. From (1), we know $W_\triangle(R|D) = W_\triangle(C|D)$, then $(x_i, x_j) \notin \mathrm{DIS}_\triangle(R|D)$. Because of arbitrary of x_i and x_j, we have $W_\triangle(R - \{a\}|D) = W_\triangle(R|D)$, which is contradictory with consequent of (2). Hence, $\Leftarrow$ holds.

From (1) and (2), the relative discernibility reduct is equivalent to Hu's discernibility matrix reduct.

Example 4 *(Continued from Example 3).* Let $\triangle = \mathrm{H}$, $R_1 = \{a\}$, and $R_2 = \{a, c\}$.

$U/C = \{\{x_1\}, \{x_2, x_3, x_4\}, \{x_5, x_6, x_7\}\}$ and $U/(D_\mathrm{H} \cup C) = \{\{x_1\}, \{x_2, x_3\}, \{x_4\}, \{x_5\}, \{x_6, x_7\}\}$, we have $\overline{W_\mathrm{H}(C)} = 19$ and $\overline{W_\mathrm{H}(D_\mathrm{H} \cup C)} = 11$, and $W_\mathrm{H}(C|D_\mathrm{H}) = 49 - 25 - 19 + 11 = 16$.

From the results of Example 3, we have $W_\mathrm{H}(R_1|D_\mathrm{H}) < W_\mathrm{H}(R_2|D_\mathrm{H}) = W_\mathrm{H}(C|D_\mathrm{H})$. Hence, R_2 is H-reduct and R_1 is not H-reduct.

In the process of the heuristic attribute reduction, the significance measure of features is one of the crucial roles, and significance measures have been widely studied [12,26,27]. Two attribute significance measures from the viewpoint of the relative discernibility are given as follows:

Definition 16. Let $\mathrm{DT}_\triangle = (U, C \cup D, V_\triangle, f_\triangle)$ be the general decision table of DT, $R \subseteq C$, and $a \in R$. The weeded significance of a in R is defined as

$$Sig_\triangle^-(a, R, D) = W_\triangle(R|D) - W_\triangle\big(R - \{a\}|D\big).$$

Definition 17. Let $\mathrm{DT}_\triangle = (U, C \cup D, V_\triangle, f_\triangle)$ be the general decision table of DT, $R \subset C$, and $a \in C - R$. The joined significance of a in R is defined as

$$Sig_\triangle^+(a, R, D) = W_\triangle\big(R \cup \{a\}|D\big) - W_\triangle(R|D).$$

Definition 18. Let $\mathrm{DT}_\triangle = (U, C \cup D, V_\triangle, f_\triangle)$ be the general decision table of DT and $a \in C$. If $Sig_\triangle^+(a, R, D) > 0$, the attribute a is a core attribute of $\mathrm{DT}_\triangle$ with respect to the relative discernibility, denoted as $a \in \mathrm{WCORE}_\triangle(C)$.

Theorem 3. $\mathrm{WCORE}_\triangle(C) = \mathrm{MCORE}_\triangle(C)$.

Proof. (1) $\Leftarrow$. From $a \in \mathrm{MCORE}_\triangle(C)$, there exist $x_i, x_j \in U (i \neq j)$, which satisfy $f_\triangle(x_i, C - \{a\}) = f_\triangle(x_j, C - \{a\}) \wedge f_\triangle(x_i, a) \neq f_\triangle(x_j, a) \wedge f_\triangle(x_i, D) \neq f_\triangle(x_j, D)$. Hence, we have $W_\triangle(C - \{a\}|D) < W_\triangle(C|D)$, that is, $a \in \mathrm{WCORE}_\triangle(C)$.

(2) $\Rightarrow$. Because of $a \in \mathrm{WCORE}_\triangle(C)$, from $W_\triangle(C - \{a\}|D) < W_\triangle(C|D)$, there exist $x_i, x_j \in U$, which satisfy $f_\triangle(x_i, a) \neq f_\triangle(x_j, a) \wedge f_\triangle(x_i, C - \{a\}) = f_\triangle(x_j, C - \{a\}) \wedge f_\triangle(x_i, D) \neq f_\triangle(x_j, D)$, that is, $a \in m_\triangle(x_i, x_j)$ and $|m_\triangle(x_i, x_j)| = 1$. Therefore, $a \in \mathrm{CORE}_\triangle(C)$.

From (1) and (2), $\mathrm{WCORE}_\triangle(C) = \mathrm{MCORE}_\triangle(C)$. $\square$

4. General attribute reduction algorithms

In this section, for the general decision table $\mathrm{DT}_\triangle$, we introduce an algorithm for computing relative discernibility $W_\triangle(R|D)$ and design two general attribute reduction algorithms based on forward search strategy and backward search strategy, which are called GARA-FS$_\triangle$ and GARA-BS$_\triangle$, respectively.

4.1. Algorithm for computing the relative discernibility $W_\triangle(R|D)$

For the general decision table $\mathrm{DT}_\triangle = (U, C \cup D, V_\triangle, f_\triangle)$ and $R \subseteq C$, according to Proposition 4 and Proposition 5, we first present an algorithm for calculating $\overline{W_\triangle(D)}$, $\overline{W_\triangle(R)}$, $\overline{W_\triangle(D \cup R)}$, and $W_\triangle(R|D)$.

Algorithm 1. For calculating $W_\triangle(R|D)$.

Input: A decision table $\mathrm{DT}_\triangle = (U, C \cup D, V_\triangle, f_\triangle)$, $R \subseteq C$;

Output: $\overline{W_\triangle(D)}$, $\overline{W_\triangle(R)}$, $\overline{W_\triangle(D \cup R)}$, and $W_\triangle(R|D)$.

Step1: Sort $\mathrm{DT}_\triangle$ by D, obtain $\mathrm{Sort}(U, D)$ and $U/D = \{X_1, X_2, \ldots, X_m\}$, and compute $\overline{W_\triangle(D)} = \sum_{j=1}^{m} |Y_j|^2$;

Step2: Sort $\mathrm{DT}_\triangle$ by R, obtain $\mathrm{Sort}(U, D \cup R)$; Let $U/R = \{Y_1, Y_2, \ldots, Y_n\}$ and compute $\overline{W_\triangle(R)} = \sum_{i=1}^{n} |X_i|^2$;

Step3: According to Step2, obtain $U/(D \cup R) = \{Z_1, Z_2, \ldots, Z_t\}$ and compute $\overline{W_\triangle(D \cup R)} = \sum_{k=1}^{t} |Z_k|^2$;

Step4: Calculate $W_\triangle(R|D) = |U|^2 - \sum_{j=1}^{m} |Y_j|^2 - \sum_{i=1}^{n} |X_i|^2 + \sum_{k=1}^{t} |Z_k|^2$;
Step5: Output $\overline{W_\triangle(D)}$, $\overline{W_\triangle(R)}$, $\overline{W_\triangle(D \cup R)}$, and $W_\triangle(R|D)$.
$\mathrm{Sort}(U, R)$ denotes the result of radix sort with regard to attributes rank R.

In this algorithm, the equivalence class partitioning plays a pivotal role. It is well known that the equivalence class U/R can be derived from comparing objects pairwise, whose time complexity is $O(|R||U|^2)$, which can be obtained by the quick sort method, whose time complexity is $O(|R||U|\log|U|)$ [18]. However, radix sort can decrease the time complexity of computing U/R to $O(|R||U|)$ [5,36]. To be consistent, the radix sort technique is adopted in all the algorithms of this study for partitioning equivalence class. Obviously, the time complexity of Algorithm 1 is $O((|R|+1)|U|)$.

4.2. General attribute reduction algorithm based on forward search strategy

Using the reduction method of discernibility matrix, we can obtain all reducts; unfortunately, NP-hardness remains a problem. The heuristics reduction methods normally can be utilized to attain an attribute subset for preserving different properties of the original decision table efficiently. From the perspective of search strategy, existing heuristic reduction methods can be roughly classified into forward selection strategy (or addition strategy) and backward elimination strategy (or deletion strategy) [39]. The forward search strategy normally begins with an empty set or the core attribute set and gradually adds features with maximum significance into the candidate attribute subset in each iteration until the stop criterion is satisfied; the backward search strategy begins with a full set of condition features and removes one-by-one redundant attributes until a reduct is constructed.

On the basis of the attribute significance and the forward greedy strategy, we first study a general forward attribute reduction algorithm based on the relative discernibility. The algorithm can be depicted as follows.

Algorithm 2. General attribute reduction algorithm based on forward search strategy (GARA-FS$_\triangle$).
Input: The general decision table $\mathrm{DT}_\triangle = (U, C \cup D, V_\triangle, f_\triangle)$;
Output: A reduct R.
Step1: $R = \mathrm{WCORE}_\triangle(C) = \Phi$;
Step2: for $i = 1$ to $|C|$ do
  if $(Sig_\triangle^-(a_i, C, D) > 0)$ $\mathrm{WCORE}_\triangle(C) = \mathrm{WCORE}_\triangle(C) \cup \{a_i\}$;
Step3: $R = \mathrm{WCORE}_\triangle(C)$ and compute $W_\triangle(R|D)$;
Step4: While $(W_\triangle(R|D) \neq W_\triangle(C|D))$
Step4.1: {for $(\forall a_k \in C - R)$, calculate $Sig_\triangle^+(a_k, R, D)$;
Step4.2: Select $a = arg_{ak} \max Sig_\triangle^+(a_k, R, D)$. If the attribute like that is not only one, select an attribute arbitrarily;
Step4.3: $R \leftarrow R \cup \{a\}$;
Step4.4: Update $W_\triangle(R|D)$;
}
Step5: Output R.

(1) Analysis of the time complexity
In Algorithm 2, the time complexity of Step2 is $O(|C|^2|U|)$ and the time complexity of Step3 is $O(|\mathrm{CORE}||U|)$. In Step4, the time complexity is $O(|C|^2|U|)$. Thus, the time complexity of Algorithm 2 is $O(|C|^2|U|)$.
(2) Analysis of the sorting number
In Algorithm 2, the sorting number of Step2 is $|C|(|C|-1)$; the sorting number of Step3 is $|\mathrm{WCORE}_\triangle(C)|$; and the sorting number of Step4 is $|R - \mathrm{CORE}_\triangle(C)| + \sum_{i=0}^{|R-\mathrm{CORE}|-1}(|C| - (|\mathrm{CORE}| + i))$. (For convenience, we let θ denote $\sum_{i=0}^{|R-\mathrm{CORE}|-1}(|C| - (|\mathrm{CORE}| + i))$, that is, $\theta = \sum_{i=0}^{|R-\mathrm{CORE}|-1}(|C| - (|\mathrm{CORE}| + i))$). Hence, the total sorting number of Algorithm 2 is $|C|^2 + |R| - |C| + \theta$.

4.3. General attribute reduction algorithm based on backward search strategy

The deletion search strategy can avoid computing the core attribute set and deterministically generate a winning reduct under a certain attributes order. The idea of the deletion strategy is that the reduct starts with the whole conditional attribute set ($R = C$), and consecutively selects an attribute (denoted as a), which relies on a certain criterion of attribute significance; if the attribute a satisfies $W_\triangle(R - \{a\}|D) = W_\triangle(C|D)$, a can be deleted from R; otherwise, a needs to be kept. When every conditional attribute has been checked as the above approach, a complete reduct can be obtained.

The order of checked conditional attributes determines the reduct result. That is, different orders of conditional attributes result in different reducts. We can construct a conditional attributes order by the diverse evaluation criterions, such as user preference, user experience, alphabetic order, and certain significance measures of conditional attributes. In this study, we do not deeply discuss how to obtain the attribute order relation, and assume that a certain relation of feature order has already existed for the decision table.

According to the above discussion, we can construct a general attribute reduction algorithm based on the backward search strategy as follows:

Algorithm 3. General attribute reduction algorithm based on backward search strategy (GARA-BS$_\triangle$).

Input: The general decision table $\mathrm{DT}_\triangle = (U, C \cup D, V_\triangle, f_\triangle)$ and an order of the conditional attributes: $SC = \{a_1, a_2, \ldots, a_{|C|}\}$;

Output: A reduct R.

Step1: $R = \Phi$, flag $=0$;

Step2: $R \leftarrow SC = \{a_1, a_2, \ldots, a_{|C|}\}$, $i = 1$;

Step3: Calculate $W_\triangle(R|D)$;

Step4: for $i = 1$ to $|C|$ do

{ Compute $W_\triangle(R - \{a_i\}|D)$;

if $(Sig^-_\triangle(a_i, C, D) = 0)$ $R = R - \{a_i\}$;

}

Step5: Output R.

(1) Analysis of the time complexity

In Algorithm 3, the time complexity of Step3 is $O(|C||U|)$; in Step4, the time complexity is $O(|C|^2|U|)$. Thus, the time complexity of Algorithm 3 is $O(|C|^2|U|)$.

(2) Analysis of the sorting number

In Algorithm 3, the sorting number of Step3 is $|C|+1$; in the worst case, the sorting number of Step4 is $|C|(1+|C|-1) = |C|^2$. Hence, the total sorting number of Algorithm 3 is $|C|^2 + |C| + 1$.

Example 5. For the decision table in Table 2, let $\triangle = \mathrm{H}$. The main steps for calculating reduct based on Algorithm 2 and Algorithm 3 are described as follows.

First, we analyze the procedures for obtaining R-reduct (i.e., H-reduct) by using Algorithm 2:

(1) Sort $\mathrm{DT_H}$ by the attribute rank D, a, b, c, d, we can get $W_\mathrm{H}(C|D) = 16$.

(2) Sort $\mathrm{DT_H}$ by the attribute rank D, b, c, d, we can obtain $W_\mathrm{H}(C - \{a\}|D) = W_\mathrm{H}(C|D) = 16$. So, $Sig^-_\mathrm{H}(a, C, D) = 0$ and $a \notin \mathrm{WCORE_H}(C)$. Similarly, we can get $Sig^-_\mathrm{H}(b, C, D) = 0$, $Sig^-_\mathrm{H}(c, C, D) = 4 > 0$, and $Sig^-_\mathrm{H}(d, C, D) = 0$, and then $b \notin \mathrm{WCORE_H}(C)$ and $c \in \mathrm{WCORE_H}(C)$. In the process of computing core attributes, the number of sort is $4 \times 4 = 16$.

(3) Let $R = \mathrm{WCORE_H}(C) = \{c\}$, we can get $\mathrm{Sort}(U, R)$ and $W_\mathrm{H}(R|D) = 12$. Sort times are 2.

(4) In the following steps, we execute Step4 of Algorithm 2.

On the basis of $\mathrm{Sort}(U, R)$, sort $\mathrm{DT_H}$ by a, we can get $W_\mathrm{H}(\{ac\}|D) = 16$ and $Sig^+_\mathrm{H}(a, R, D) = 4$. Sort times are 2.

On the basis of $\mathrm{Sort}(U, R)$, sort $\mathrm{DT_H}$ by b, we can get $W_\mathrm{H}(\{bc\}|D) = 16$ and $Sig^+_\mathrm{H}(b, R, D) = 4$. Sort times are 2.

On the basis of $\mathrm{Sort}(U, R)$, sort $\mathrm{DT_H}$ by d, we can get $W_\mathrm{H}(\{cd\}|D) = 16$ and $Sig^+_\mathrm{H}(d, R, D) = 0$. Sort times are 2.

Select a into R and $R = \{ac\}$ and sort $\mathrm{DT_H}$ by the attribute set R, we can get $W_\mathrm{H}(R|D) = W_\mathrm{H}(C|D) = 16$. Then, the loop is over.

(5) We get a reduct $R = \{ac\}$ and the total number of sort is $16 + 2 + 2 + 2 + 2 = 24$.

Then, we analyze the procedures for obtaining R-reduct (i.e., H-reduct) by using Algorithm 3 as follows.

(1) Let SC: $a \prec b \prec c \prec d$. Sort $\mathrm{DT_H}$ by the attribute rank D, a, b, c, d, we can obtain $W_\mathrm{H}(C|D_\mathrm{H}) = 16$. The sorting number of this step is 5.

(2) In the following steps, we execute Step4 of Algorithm 3.

Sort $\mathrm{DT_H}$ by the attribute rank D, b, c, d, we can obtain $W_\mathrm{H}(SC - \{a\}|D_\mathrm{H}) = W_\mathrm{H}(C|D_\mathrm{H}) = 16$ and a can be deleted. Then SC: $b \prec c \prec d$ and sort times are 4.

Sort $\mathrm{DT_H}$ by the attribute rank D, c, d, we can obtain $W_\mathrm{H}(SC - \{b\}|D) = 10 < W_\mathrm{H}(C|D_\mathrm{H})$ and b cannot be deleted. Then SC: $b \prec c \prec d$ and sort times are 3.

Sort $\mathrm{DT_H}$ by the attribute rank D, b, d, we can obtain $W_\mathrm{H}(SC - \{c\}|D) = 6 < W_\mathrm{H}(C|D_\mathrm{H})$ and c cannot be deleted. Then SC: $b \prec c \prec d$ and sort times are 3.

Sort $\mathrm{DT_H}$ by the attribute rank D, b, c, we can obtain $W_\mathrm{H}(SC - \{d\}|D_\mathrm{H}) = 16 = W_\mathrm{H}(C|D_\mathrm{H})$ and d can be deleted. Sort times are 3.

(3) Finally, the loop ends. We can obtain a reduct $R = \{bc\}$ and the total number of sort is $5 + 4 + 3 + 3 + 3 = 18$.

5. Quick general attribute reduction algorithm

To improve efficiency of two reduction algorithms GARA-FS$_\triangle$ and GARA-BS$_\triangle$, in this section, we give an acceleration approach through reducing the radix sort times to improve performance of the aforementioned two reduction algorithms. First, the acceleration strategy and properties for improving the performance of algorithms are introduced. Then, on the basis of acceleration approach, a quick algorithm for calculating core attributes is proposed, and two quick general attribute reduction algorithms based on forward addition strategy (QGARA-FS$_\triangle$) and backward deletion strategy (QGARA-BS$_\triangle$) are designed.

5.1. Strategy of improving the algorithm performance

Computing equivalence class is pivotal to the reduction in inconsistent decision tables. Some scholars have presented many approaches to quickly achieve the equivalence class division. For example, Liu et al. [18] presented the quick sort to partition the equivalence class; Xu et al. [36] and Ge et al. [5] presented the radix sort to partition the equivalence class; Liu et al. [19] proposed the hash sort to compute equivalence class.

Following the algorithms of Section 4, we also make use of the radix sort to calculate equivalence class. Furthermore, for high-dimensional data sets, the above approaches for computing equivalence class in the process of attribute reduction are still inefficient. If we can drastically decrease the number of radix sort of the equivalence class computation in the reduction procession, one can significantly improve the performance of reduction algorithms. On the basis of the above idea, we present a property (described as Proposition 6) that reveals the radix sort discipline that exists in the complete decision systems. In this nature, we should arrive at the purpose for accelerating the reduction algorithms.

For a decision table $\mathrm{DT}_\triangle = (U, C \cup D, V_\triangle, f_\triangle)$, in the process of computing $U/C - \{c_i\}$ and $U/C - \{c_{i+1}\}$, we can find that there exist $|C| - 2$ times repetitive sorts in sorting process $c_{i+2}, \ldots, c_{|C|}, c_1, \ldots, c_{i-1}$, and has only once different sort. Because the radix sort is stable, we can obtain $U/(C - \{c_{i+1}\})$ derived from $U/(C - \{c_i\})$ [26,44].

Proposition 6. *For a decision table* $\mathrm{DT}_\triangle = (U, C \cup D, V_\triangle, f_\triangle)$, $\mathrm{Sort}(U, C - \{c_{i+1}\}) = \mathrm{Sort}(\mathrm{Sort}(U, C - \{c_i\}), \{c_i\})$.

Proof. $\mathrm{Sort}(U, C - \{c_{i+1}\})$ denotes the radix sorting result with respect to attributes $c_{i+2}, \ldots, c_{|C|}, c_1, \ldots, c_i$; we divide two steps to obtain $\mathrm{Sort}(\mathrm{Sort}(U, C - \{c_i\}), \{c_i\})$. That is, first, we can obtain $\mathrm{Sort}(U, C - \{c_i\})$ by radix sort in terms of attributes $c_{i+1}, \ldots, c_{|C|}, c_1, \ldots, c_{i-1}$; then, based on $\mathrm{Sort}(U, C - \{c_i\})$, we complete radix sort according to attribute c_i. Therefore, the radix sorting result with regard to attributes $c_{i+1}, c_{i+2}, \ldots, c_{|C|}, c_1, \ldots, c_i$ is equivalent to merge the above steps. Because the radix sort is stable, $\mathrm{Sort}(U, C - \{c_{i+1}\})$ equals $\mathrm{Sort}(\mathrm{Sort}(U, C - \{c_i\}), \{c_i\})$. Thus, the proposition holds.

Hence, Proposition 6 can be applied to calculate the core attribute set and delete redundant features from the attribute set to increase efficiency of reduction algorithms. □

5.2. Quick calculating core attribute algorithm

Computing core attributes is a crucial step in the forward search strategy reduction method. In Algorithm 2, the time complexity of calculating core attributes is $O(|C|^2|U|)$, while the radix sorting number of computing core attributes is $|C|(|C| - 1)$, which is larger for high-dimensional information systems. According to Definition 18 and Proposition 6, a fast core attribute computing algorithm whose performance is improved by decreasing the radix sort times is given as follows:

Algorithm 4. Quick calculating core attributes algorithm (QCCA).

Input: The general decision table $\mathrm{DT}_\triangle = (U, C \cup D, V_\triangle, f_\triangle)$;

Output: $\mathrm{WCORE}_\triangle(C)$.

Step1: Sort $\mathrm{DT}_\triangle$ by $C \cup D$, we can obtain $\mathrm{Sort}(U, C - \{c_1\})$, $\mathrm{Sort}(U, C)$, $\mathrm{Sort}(U, C - \{c_1\} \cup D)$, and $\mathrm{Sort}(U, C \cup D)$, and then compute $W_\triangle(C|D)$ and $W_\triangle(C - \{c_1\}|D)$;

Step2: if $(Sig^-_\triangle(c_1, C, D) > 0)$ $\mathrm{WCORE}_\triangle(C) = \mathrm{WCORE}_\triangle(C) \cup \{c_1\}$;

Step3: for $i = 1$ to $|C| - 1$ do

Step3.1: { Based on $\mathrm{Sort}(U, C - \{c_{i-1}\})$, sort $\mathrm{DT}_\triangle$ by the attribute c_i for obtaining $\mathrm{Sort}(U, C - \{c_{i+1}\})$;

// if $(i - 1 = 0)$, then c_{i-1} is an empty set.

Step3.2: Subsequently, sort $\mathrm{DT}_\triangle$ by D to attain $\mathrm{Sort}(U, D \cup C - \{c_{i+1}\})$ and $W_\triangle(C - \{c_{i+1}\}|D)$;

Step3.3: If $(Sig^-_\triangle(c_{i+1}, C, D) > 0)$ $\mathrm{WCORE}_\triangle(C) = \mathrm{WCORE}_\triangle(C) \cup \{c_{i+1}\}$;

}

Step4: Output $\mathrm{WCORE}_\triangle(C)$.

(1) Analysis of the time complexity

In Algorithm 4, from Step1 to Step2, the time complexity is $O(|C||U|)$; The time complexities of Step3.1 and Step3.2 are $O(|U|)$ and $O(|C||U|)$, respectively. As the iterations time of Step3 is $|C| - 1$, the time complexity of Step3 is $O(|C|^2|U|)$. Thus, the time complexity of Algorithm 4 is $O(|C|^2|U|)$.

(2) The analysis of the number of sort

In Step1, the number of sort is $|C| + 1$; the sorting number of Step3 is $2(|C| - 1)$. Thus, the sorting number of Algorithm 4 is $3|C| - 1$. In ref. [27], to compute core attributes, we should compute $U/(C - \{a\})$ for every $a \in C$. Hence, its sorting number is $|C|(|C| - 1)$.

Example 6. For the decision table in Table 2, let $\triangle = \mathrm{H}$. The process of calculating core attributes through Algorithm 4 is described as follows:

(1) Sort $\mathrm{DT_H}$ by the attribute rank a, b, c, d, D to obtain $\mathrm{Sort}(U, C - \{a\})$, $\mathrm{Sort}(U, C)$, $\mathrm{Sort}(U, D \cup C - \{a\})$, and $\mathrm{Sort}(U, C \cup D)$, and one can get $W_\mathrm{H}(C|D) = 16$ and $W_\mathrm{H}(C - \{a\}|D) = 16$. Because of $W_\mathrm{H}(C - \{a\}|D) = W_\mathrm{H}(C|D)$, $a \notin \mathrm{WCORE_H}(C)$ and the number of sort is 5.

(2) In the following steps, we execute Step3 of Algorithm 4.

On the basis of Sort(U, C), sort $\mathrm{DT_H}$ by a, we can obtain Sort$(U, C - \{b\})$; subsequently, sort $\mathrm{DT}_\triangle$ by D to attain Sort$(U, C - \{b\} \cup D)$ and $W_\mathrm{H}(C - \{b\}|D) = 16$; so $b \notin \mathrm{WCORE_H}(C)$ and the number of sort is 2.

On the basis of Sort$(U, C - \{b\})$, sort $\mathrm{DT_H}$ by b, we can obtain Sort$(U, C - \{c\})$; subsequently, sort $\mathrm{DT}_\triangle$ by D to attain Sort$(U, C - \{c\} \cup D)$ and $W_\mathrm{H}(C - \{c\}|D) = 6$; so $c \in \mathrm{WCORE_H}(C)$ and the number of sort is 2.

On the basis of Sort$(U, C - \{c\})$, sort $\mathrm{DT_H}$ by c, we can obtain Sort$(U, C - \{d\})$; subsequently, sort $\mathrm{DT}_\triangle$ by D to attain Sort$(U, C - \{d\} \cup D)$ and $W_\mathrm{H}(C - \{d\}|D) = 16$; $d \notin \mathrm{WCORE_H}(C)$ and the number of sort is 2.

(3) Hence, $\mathrm{WCORE_H}(C) = \{c\}$ and the number of sort is $5 + 2 + 2 + 2 = 11$.

5.3. Quick general attribute reduction algorithms

From the research in the previous subsections, first, a quick forward attribute reduction algorithm based on the relative discernibility is depicted as follows:

Algorithm 5. Quick general attribute reduction algorithm based on forward search strategy (QGARA-FS$_\triangle$).

Input: The general decision table $\mathrm{DT}_\triangle = (U, C \cup D, V_\triangle, f_\triangle)$;

Output: A reduct R.

Step1: $R = \Phi$, flag $= 0$;

Step2: According to Algorithm 4, attain the attribute core set $\mathrm{WCORE}_\triangle(C)$;

Step3: $R = \mathrm{WCORE}_\triangle(C)$, compute $W_\triangle(R|D)$;

Step4: While $(W_\triangle(R|D) \neq W_\triangle(C|D))$

Step4.1: {for ($\forall a_k \in C - R$), calculate $Sig_\triangle^+(a_k, R, D)$;

Select $a = arg_{ak} \max Sig_\triangle^+(a_k, R, D)$; if the attribute like that is not only one, select an attribute arbitrarily;

Step4.2: $R \leftarrow R \cup \{a\}$;

Step4.3: Update $W_\triangle(R|D)$;

}

Step5: Output R.

(1) Analysis and comparison of the time complexity

In Algorithm 5, the time complexity of Step2 is $O(|C|^2|U|)$ and the time complexity of Step3 is $O(|\mathrm{CORE}_\triangle(C)||U|)$; In Step4, the time complexity is $O(|C|^2|U|)$. Thus, the time complexity of Algorithm 5 is $O(|C|^2|U|)$.

(2) Analysis of the sorting number

In Algorithm 5, the sorting number of Step2 is $2|C| - 1$; the sorting number of Step3 is $|\mathrm{CORE}_\triangle(C)|$; and the sorting number of Step4 is $|R - \mathrm{CORE}_\triangle(C)| + \theta$. Hence, the total sorting number of Algorithm 5 is $2|C| + |R| - 1 + \theta$.

Example 7. For the decision table in Table 1, let $\triangle = \mathrm{H}$, the main steps for calculating attribute reduction based on Algorithm 5 is described as follows:

(1) First, from Example 6, we have $\mathrm{WCORE_H}(C) = \{c\}$. The number of sort is 11.

(2) Let $R = \mathrm{WCORE_H}(C) = \{c\}$. Sort $\mathrm{DT_H}$ by the attribute rank R, D, we can get Sort(U, R) and Sort$(U, R \cup D)$ and $W_\mathrm{H}(R|D) = 12$. Sort times are 2.

(3) In the following steps, we execute Step4 of Algorithm 5.

On the basis of Sort(U, R), sort $\mathrm{DT_H}$ by the attribute rank b, D, we can get $W_\mathrm{H}(\{ac\}|D) = 16$ and $Sig_\mathrm{H}^+(a, R, D) = 4$. Sort times are 2.

On the basis of Sort(U, R), sort $\mathrm{DT_H}$ by the attribute rank a, D, we can get $W_\mathrm{H}(\{bc\}|D) = 16$ and $Sig_\mathrm{H}^+(b, R, D) = 4$. Sort times are 2.

On the basis of Sort(U, R), sort $\mathrm{DT_H}$ by the attribute rank d, D, we can get $W_\mathrm{H}(\{cd\}|D) = 16$ and $Sig_\mathrm{H}^+(b, R, D) = 0$. Sort times are 2.

Hence, a is selected into R, and then $R = \{ac\}$. On the basis of Sort(U, R), sort $\mathrm{DT_H}$ by the attribute rank a, D, one can get $W_\mathrm{H}(R|D) = 16 = W_\mathrm{H}(C|D)$. The loop ends and sort times are 2.

(4) Finally, the reduct $R = \{ac\}$ and the number of sort is $11 + 2 + 2 + 2 + 2 + 2 = 21$.

Proposition 7. *Given the general decision table $\mathrm{DT}_\triangle = (U, C \cup D, V_\triangle, f_\triangle)$, if a conditional attribute subset R of C satisfies that $\forall a \in R$, $W_\triangle(R - \{a\}|D) \neq W_\triangle(C|D)$, then R is a real reduct.*

Proof. Suppose R is not a reduct. There exists a redundant attribute, denoted as b, which meets $W_\triangle(R - \{b\}|D) = W_\triangle(C|D)$. From Definition 15, b can be deleted. Hence, the assumption does not hold and Proposition 7 holds.

According to Proposition 7, let a conditional attribute order be SC: $a_1 \leq a_2 \leq \ldots \leq a_{|C|}$, we select $a_i \in SC$ $(1 \leq i \leq |C|)$ in turn. If a_i meets $W_\triangle(SC - \{a_i\}|D) = W_\triangle(C|D)$, then a_i should be deleted from SC; otherwise, a_i cannot be deleted from SC. The operation is repeated for all attributes of SC, and let the remaining attributes in SC be R. Each attribute of R is independent and R is a real reduct. □

Proposition 8. *Let $R \subseteq C$, there exists the sort result of* Sort$(U, R-\{a_i\})$*, we can partition $U/(R-\{a_i\})$ and calculate $W_\triangle(R-\{a_i\}|D)$. If $\exists a_i \in R$, $W_\triangle(R|D) = W_\triangle(R-\{a_i\}|D)$, the attribute a_i is a redundant attribute and can be deleted. And then, we no longer need to sort by a_i to get $W_\triangle((R-\{a_i\}-\{a_{i+1}\})|D)$, only need to partition $U/(R-\{a_i\}-\{a_{i+1}\})$ and calculate $W_\triangle((R-\{a_i\}-\{a_{i+1}\})|D)$ based on the result of* Sort$(U, R-\{a_i\})$.

From Proposition 6, Proposition 8 holds.

According to the above discussion, we propose a general attribute reduction algorithm based on backward search strategy as follows.

Algorithm 6. Quick general attribute reduction algorithm based on backward search strategy (QGARA-BS$_\triangle$).

Input: The general decision table $\mathrm{DT}_\triangle = (U, C \cup D, V_\triangle, f_\triangle)$ and an order of the conditional attributes: $SC = \{a_1, a_2, \ldots, a_{|C|}\}$;

Output: A reduct R.

Step1: $R = \Phi$, flag $= 0$;

Step2: $R \leftarrow SC = \{a_1, a_2, \ldots, a_{|C|}\}$, $i = 1$;

Step3: In the process of sort $\mathrm{DT}_\triangle$ by $R \cup D$, we can obtain Sort(U, R) and Sort$(U, R \cup D)$ and compute $W_\triangle(R|D)$ and $W_\triangle(R-\{a_1\}|D)$;

Step4: if ($Sig^-_\triangle(a_1, R, D) = 0$) { $R = R - \{a_1\}$; flag $= 1$; }

Step5: for $i = 1$ to $|R| - 1$ do

Step5.1: { if (flag) // a_i can be deleted

{ Sort $\mathrm{DT}_\triangle$ by D to attain Sort$(U, R-\{a_{i+1}\} \cup D)$ and $W_\triangle(R-\{a_{i+1}\}|D)$;

Compute $W_\triangle(R-\{a_{i+1}\}|D)$; }

else // a_i cannot be deleted

{ Sort $\mathrm{DT}_\triangle$ by the attribute a_i based on Sort$(U, R-\{a_i\})$ for obtaining Sort$(U, R-\{a_{i+1}\})$;

Subsequently, sort $\mathrm{DT}_\triangle$ by D to attain Sort$(U, R-\{a_{i+1}\} \cup D)$;

Compute $W_\triangle(R-\{a_{i+1}\}|D)$;

}

Step5.2: if ($Sig^-_\triangle(a_{i+1}, R, D) = 0$) { $R = R - \{a_{i+1}\}$; flag $= 1$;}

else flag $= 0$;

}

Step6: Output R.

(1) Analysis and comparison of the time complexity

In Algorithm 6, the time complexities of Step3 and Step4 are both $O(|C||U|)$; in Step5, the time complexity is $O(|C|^2|U|)$. Thus, the time complexity of Algorithm 6 is $O(|C|^2|U|)$.

(2) Analysis and comparison of the sorting number

In Algorithm 6, the sorting numbers of Step3 and Step5 are $|C| + 1$ and $|C| - 1$, respectively. Hence, the total sorting number of Algorithm 6 is $2|C| + 1$.

Example 8. For the decision table in Table 2, let $\triangle = \mathrm{H}$, the main steps for calculating reduct based on Algorithm 6 is described as follows.

(1) First, let SC: $a \prec b \prec c \prec d$ and $R \leftarrow SC$. Sort $\mathrm{DT_H}$ by the attribute rank a, b, c, d, D, we can obtain Sort$(U, R-\{a\})$, Sort(U, R), Sort$(U, R-\{a\} \cup D)$, and Sort$(U, R \cup D)$, and also can get $W_\mathrm{H}(R-\{a\}|D) = 16$ and $W_\mathrm{H}(R|D) = 16$, such that a can be deleted. Then, R: $b \prec c \prec d$. Sort times are 5.

(2) In the following steps, we execute Step5 of Algorithm 6.

On the basis of Sort(U, R), we can obtain Sort$(U, R-\{b\})$. One sort $\mathrm{DT}_\triangle$ by D for attaining Sort$(U, R-\{b\} \cup D)$ and get $W_\mathrm{H}(R-\{b\}|D) = 12$, such that b cannot be deleted. Then, R: $b \prec c \prec d$. Sort times are 1.

On the basis of Sort$(U, R-\{b\})$, sort $\mathrm{DT_H}$ by b, we can obtain Sort$(U, R-\{c\})$; Subsequently, sort $\mathrm{DT}_\triangle$ by D for attaining Sort$(U, R-\{c\} \cup D)$, we can get $W_\mathrm{H}(R-\{c\}|D) = 12$, such that c cannot be deleted. Then, R: $b \prec c \prec d$. Sort times are 2.

On the basis of Sort$(U, R-\{c\})$, sort $\mathrm{DT_H}$ by c, we can obtain Sort$(U, R-\{d\})$; Subsequently, sort $\mathrm{DT}_\triangle$ by D for attaining Sort$(U, R-\{d\} \cup D)$ and get $W_\mathrm{H}(R-\{d\}|D) = 16$, such that d can be deleted. Then, R: $b \prec c$. Sort times are 2.

(3) Finally, we can obtain a reduct $R = \{bc\}$ and the total number of sort is $5 + 1 + 2 + 2 = 10$.

Remark. For Table 2, comparing with the results of reducts, GARA-FS$_\mathrm{H}$ and QGARA-FS$_\mathrm{H}$ have the same result which is $R = \{ac\}$, and GARA-BS$_\mathrm{H}$ and QGARA-BS$_\mathrm{H}$ have the same result, which is $R = \{bc\}$. Comparing with the sort times, the sorting number of four reduction algorithms differs from each other, namely GARA-FS$_\mathrm{H}$ is 24, GARA-BS$_\mathrm{H}$ is 18, QGARA-FS$_\mathrm{H}$ is 21, and QGARA-BS$_\mathrm{H}$ is only 10.

In Table 3, we present comparisons of the sorting number for four reduction algorithms (GARA-FS△, GARA-BS△, QGARA-FS$_\triangle$, and QGARA-BS$_\triangle$).

Table 3
Comparison of sorting number among four attribute reduction algorithms.

Operations	GARA		QGARA	
	GARA-FS$_\triangle$	GARA-BS$_\triangle$	QGARA-FS$_\triangle$	QGARA-BS$_\triangle$
(1) Compute core	$\lvert C\rvert(\lvert C\rvert-1)$	/	$2\lvert C\rvert-1$	/
(2) Other steps	$\lvert CORE\rvert$	$\lvert C\rvert+1$	$\lvert CORE\rvert$	$\lvert C\rvert+2$
(3) Iteration structure	$\lvert CORE\rvert+\lvert R-CORE\rvert+\theta$	$\lvert C\rvert^2$	$\lvert CORE\rvert+\lvert R-CORE\rvert+\theta$	$\lvert C\rvert-1$
Total	$\lvert C\rvert^2+\lvert R\rvert-\lvert C\rvert+\theta$	$\lvert C\rvert^2+\lvert C\rvert+1$	$3\lvert C\rvert+2\lvert R\rvert-3+\theta$	$2\lvert C\rvert+1$

Table 4
Descriptions of 12 data sets.

ID	Data set	# of objects	# of attributes	# of core attributes	# of classes
1	Chess	3196	36	27	2
2	Landsat-trn	4435	37	0	7
3	Gene	3190	60	0	3
4	Semeion	1593	256	0	10
5	Mushroom	8124	22	0	2
6	Nursery	12,960	8	8	5
7	Shuttle	58,000	9	1	7
8	Handwritten	5620	64	0	10
9	Ticdata2000	5822	85	9	2
10	Connect-4	67,557	42	0	3
11	CNAE-9	1080	856	17	9

In Table 3, "/" denotes that algorithms do not exist in this step of operation and $\theta=\sum_{i=0}^{|R-CORE|-1}(|C|-(|CORE|+i))$.

From Table 3, we can find that the sorting number of QGARA-FS$_\triangle$ is less than one of GARA-FS$_\triangle$ in the step of computing core, and the number of sort is the same as in other steps. In addition, the sort time of QGARA-BS$_\triangle$ is lower than that of GARA-BS$_\triangle$ in the iteration step. As a whole, the sort time of QGARA-BS$_\triangle$ is lower than the sorting number of other three algorithms.

6. Experiments

In this section, we present a series of experiments to show the effectiveness and efficiency for the proposed reduction algorithms. The experiment data use 11 UCI data sets (https://archive.ics.uci.edu/ml/datasets.html) whose basic information is outlined in Table 4. All the experiments were carried out on a personal computer with Windows XP, Pentium Dual-CORE CPU i5-3470 3.20 GHZ and 3.47 GB memory. The software used was Microsoft Visual Studio 2005, and the programming language was C++.

To illustrate the feasibility, efficiency, and reduct results of the proposed four algorithms (GARA-FS$_\triangle$, GARA-BS$_\triangle$, QGARA-FS$_\triangle$, and QGARA-BS$_\triangle$), the experiments are divided into four parts. In the first part, the efficiency is illustrated mainly through comparing the proposed reduction algorithms with the usual and representative reduction algorithms for large data sets of UCI. To further compare the efficiency of the proposed four reduction algorithms, two incremental experiments are conducted in the second and third parts. In the last part, we compare the reduct results of two reduction strategies, that is, the forward search strategy and the backward search strategy.

Under the framework of the general reduction algorithm, as approaches and results of five types of reducts are similar, we mainly take H-reduct, for example, to illustrate the efficiency and feasibility of the proposed reduction algorithms from the second to fourth part.

6.1. Efficiency comparison of the proposed four reduction algorithms for five types of typical reducts

In this subsection, to illustrate effectiveness and efficiency of the proposed four reduction algorithms (GARA-FS$_\triangle$, GARA-BS$_\triangle$, QGARA-FS$_\triangle$, and QGARA-BS$_\triangle$) for five types of attribute reducts (H-reduct, P-reduct, D-reduct, Md-reduct, and A-reduct), we compare four attribute reduction algorithms with discernibility matrix-based reduction approaches and other corresponding representative reduction methods. Experimental results are shown in Tables 5–10, where H-matrix denotes the reduction method of Hu's discernibility matrix [6], Y-matrix represents the reduction approach of Yang's discernibility matrix [37], Z-matrix denotes the reduction approach of Zhang's discernibility matrix [41], SortN represents the number of sort, Time denotes the time consumed by the corresponding reduction algorithms, and "–" denotes that the data set is too large to achieve reduction algorithm.

From Tables 5–10, we can observe the following results:

(1) The proposed four reduction algorithms for data sets Chess, Landsat-trn, Gene, and Semeion are quicker than the corresponding discernibility matrix-based reduction algorithms. However, other data sets cannot be used for the discerni-

Table 5
Performance comparison of discernibility relation reduct algorithms.

Data set	H-Matrix/s	GARA-FS$_H$		GARA-BS$_H$		QGARA-FS$_H$		QGARA-BS$_H$	
		SortN	Time/s	SortN	Time/s	SortN	Time/s	SortN	Time/s
Chess	4.091	1379	0.595	1137	0.465	117	0.072	65	0.043
Landsat-trn	16.216	1571	0.720	704	0.337	309	0.250	44	0.042
Gene	18.537	4226	1.949	1914	0.923	684	0.627	71	0.074
Semeion	29.500	70,266	32.079	34,463	19.549	4984	7.157	301	0.393
Mushroom	–	593	0.491	274	0.248	129	0.166	26	0.046
Nursery	–	82	0.074	65	0.062	24	0.029	16	0.021
Shuttle	–	116	0.546	58	0.293	42	0.281	13	0.106
Handwritten	–	4595	3.401	2158	1.751	561	0.906	74	0.128
Ticdata2000	–	8307	9.955	4314	4.256	1165	2.108	109	0.230
Connect-4	–	2725	43.946	1724	21.840	1001	21.730	83	1.647
CNAE-9	–	781,459	1374.580	393,889	660.214	49,577	196.112	952	3.814

Table 6
Performance comparison of positive-region reduct algorithms.

Data set	Y-Matrix/s	FSPA-PR		GARA-FS$_P$		GARA-BS$_P$		QGARA-FS$_P$		QGARA-BS$_P$	
		SortN	Time/s	SortN	Time/s	SortN	Time/s	SortN	Time/s	SortN	Time/s
Chess	4.101	1662	0.254	1379	0.572	1137	0.445	117	0.069	65	0.041
Landsat-trn	16.145	1844	0.417	1571	0.737	704	0.330	309	0.270	44	0.041
Gene	18.359	6110	1.359	4226	1.966	1914	0.931	684	0.630	71	0.080
Semeion	29.593	126,785	31.086	70,266	32.527	34,463	19.487	4984	7.133	301	0.375
Mushroom	–	699	0.229	593	0.521	274	0.262	129	0.165	26	0.047
Nursery	–	72	0.029	82	0.073	65	0.063	24	0.030	16	0.021
Shuttle	–	123	0.278	116	0.548	58	0.295	42	0.285	13	0.106
Handwritten	–	5800	2.409	4595	3.420	2158	1.766	561	0.907	74	0.131
Ticdata2000	–	22,463	10.149	8307	9.998	4314	4.229	1165	2.126	109	0.232
Connect-4	–	13,839	108.082	2725	44.343	1724	21.884	1001	21.770	83	1.662
CNAE-9	–	3,397,537	2557.479	781,459	1436.687	393,889	716.282	49,577	189.301	952	4.198

Table 7
Performance comparison of distribution reduct algorithms.

Data set	Z-Matrix/s	FSPA-SCE		GARA-FS$_D$		GARA-BS$_D$		QGARA-FS$_D$		QGARA-BS$_D$	
		SortN	Time/s	SortN	Time/s	SortN	Time/s	SortN	Time/s	SortN	Time/s
Chess	4.113	1394	0.284	1379	0.581	1137	0.453	117	0.076	65	0.048
Landsat-trn	16.202	1630	0.567	1571	0.737	704	0.357	309	0.270	44	0.059
Gene	18.352	5872	1.537	4226	1.966	1914	0.931	684	0.630	71	0.080
Semeion	29.474	77,831	34.277	70,266	31.969	34,463	19.547	4984	7.278	301	0.386
Mushroom	–	705	0.466	593	0.535	274	0.277	129	0.178	26	0.061
Nursery	–	67	0.051	82	0.120	65	0.078	24	0.031	16	0.069
Shuttle	–	116	0.431	116	0.772	58	0.515	42	0.301	13	0.325
Handwritten	–	4980	3.148	4595	3.464	2158	1.801	561	0.941	74	0.165
Ticdata2000	–	7397	4.033	8307	9.945	4314	4.170	1165	2.109	109	0.249
Connect-4	–	2747	36.346	2725	44.139	1724	21.986	1001	21.920	83	1.790
CNAE-9	–	1,057,669	1974.852	781,459	1509.439	393,889	687.876	49,577	191.161	952	4.196

bility matrix-based reduction method, because these data sets are larger to result in memory overflow, while our proposed reduction algorithms can successfully complete attribute reduction.

(2) The computational time of the proposed four reduction algorithms follows the order: GARA-FS$_\triangle$ > GARA-BS$_\triangle$, QGARA-FS$_\triangle$ > QGARA-BS$_\triangle$, and QGARA-FS$_\triangle$ > QGARA-BS$_\triangle$; the consuming time of QGARA-BS$_\triangle$ is the smallest among the four algorithms. The phenomena mainly stem from different sort times. Namely, the sorting number of GARA-FS$_\triangle$ is larger than that of GARA-BS$_\triangle$. Similarly, the sorting number of QGARA-FS$_\triangle$ is larger than that of QGARA-BS$_\triangle$. The sort time of QGARA-BS$_\triangle$ is the smallest of the four algorithms.

For example, in Table 5, for the Ticdata2000 data set, the consuming times of GARA-FS$_H$, GARA-BS$_H$, and QGARA-FS$_H$ are 9.9551, 4.2563, and 2.1086 s, respectively. The computational time of QGARA-BS$_H$ is only 0.2302 s. This is because the sorting numbers of GARA-FS$_H$, GARA-BS$_H$, and QGARA-FS$_H$ are 8307, 4314, and 1165, respectively. The sort time of QGARA-BS$_H$ is only 109, which is 1/70 that of GARA-FS$_H$, 1/40 that of GARA-BS$_H$, and 1/10 that of QGARA-FS$_H$. Similar principles exist in Tables 6–10.

(3) Qian et al. [27] put forward to an accelerated framework to accelerate positive-region reduct (FSPA-PR), Shannon's entropy reduct (FSPA-SCE), complementary entropy reduct (FSPA-LCE), and combination entropy reduct (FSPA-CCE).

Table 8
Performance comparison of maximum distribution reduct algorithms.

Data set	Z-Matrix/s	GARA-FS$_{MD}$		GARA-BS$_{MD}$		QGARA-FS$_{MD}$		QGARA-BS$_{MD}$	
		SortN	Time/s	SortN	Time/s	SortN	Time/s	SortN	Time/s
Chess	4.103	1379	0.583	1137	0.457	117	0.075	65	0.047
Landsat-trn	16.199	1571	0.713	704	0.347	309	0.301	44	0.059
Gene	18.365	4226	1.949	1914	0.923	684	0.627	71	0.074
Semeion	29.592	70,266	32.455	34,463	19.644	4984	7.225	301	0.390
Mushroom	–	593	0.533	274	0.276	129	0.180	26	0.060
Nursery	–	82	0.122	65	0.111	24	0.080	16	0.069
Shuttle	–	116	0.774	58	0.521	42	0.505	13	0.329
Handwritten	–	4595	3.492	2158	1.802	561	0.944	74	0.166
Ticdata2000	–	8307	10.088	4314	4.254	1165	2.109	109	0.250
Connect-4	–	2725	44.148	1724	22.209	1001	21.979	83	1.786
CNAE-9	–	781,459	1436.687	393,889	687.876	49,577	189.301	952	4.226

Table 9
Performance comparison of assignment reduct algorithms.

Data set	Z-Matrix/s	GARA-FS$_{A}$		GARA-BS$_{A}$		QGARA-FS$_{A}$		QGARA-BS$_{A}$	
		SortN	Time/s	SortN	Time/s	SortN	Time/s	SortN	Time/s
Chess	4.112	1379	0.584	1137	0.457	117	0.081	65	0.052
Landsat-trn	16.280	1571	0.722	704	0.355	309	0.270	44	0.065
Gene	18.379	4226	1.967	1914	0.936	684	0.641	71	0.079
Semeion	29.587	70,266	32.043	34,463	19.439	4984	7.166	301	0.386
Mushroom	–	593	0.543	274	0.288	129	0.189	26	0.070
Nursery	–	82	0.140	65	0.130	24	0.095	16	0.089
Shuttle	–	116	0.864	58	0.601	42	0.586	13	0.408
Handwritten	–	4595	3.480	2158	1.824	561	0.946	74	0.174
Ticdata2000	–	8307	10.009	4314	4.218	1165	2.123	109	0.256
Connect-4	–	2725	44.377	1724	22.102	1001	21.945	83	1.882
CNAE-9	–	78,1459	1459.129	393,889	691.876	49,577	191.291	952	4.813

Table 10
Number of selected features for 10 parts.

Data set	1	2	3	4	5	6	7	8	9	10
Mushroom	2	4	7	9	11	13	15	18	20	22
Chess	4	7	11	14	18	22	25	29	32	36
Landsat-trn	4	7	11	14	18	22	25	29	32	37
Connect-4	4	8	13	17	21	25	29	34	38	42
Gene	6	12	18	24	30	36	42	48	54	60
Ticdata2000	6	13	19	26	32	38	45	51	58	64
Handwritten	9	17	26	34	43	51	60	68	77	85
Semeion	26	51	77	102	128	154	179	205	230	256
CNAE-9	86	171	257	342	428	514	599	685	770	856

In Table 6, we compare the proposed four positive-region algorithms (GARA-FS$_P$, GARA-BS$_P$, QGARA-FS$_P$, and QGARA-BS$_P$) with FSPA-PR. The efficiencies of QARA-FS$_P$ and QARA-BS$_P$ are both higher than that of FSPA-PR evidently. The main reason is that the sort times of QARA-FS$_P$ and QARA-BS$_P$ are both smaller than that of FSPA-PR. For example, for Connect-4 data set, the computational times of QARA-FS$_P$ and QARA-BS$_P$ are 21.770 and 1.662 s, which are 1/5 and 1/65 that of FSPA-PR, respectively. The sort times of FSPA-PR, QARA-FS$_P$, and QARA-BS$_P$ are 13,839, 1001, and 83, respectively.

Shannon's entropy reduct is equivalent to the distribution reduct [7]. In Table 7, we additionally compare the proposed four distribution reduction algorithms (GARA-FS$_D$, GARA-BS$_D$, QGARA-FS$_D$, and QGARA-BS$_D$) with FSPA-SCE [27]. Similar to Table 6, the principles of Table 6 also exist in Table 7.

6.2. Comparison of reduction algorithms for different proportion data sets

In this subsection, we evaluate the time consumption of the proposed four algorithms with the increase in the size of data in terms of computing R-reduct (i.e., H-reduct) on nine data sets, that is, Nursery, Shuttle, Mushroom, Landsat-trn, Connect-4, Gene, Handwritten, Semeion, and CNAE-9.

We divide each of these nine data sets into 10 parts of equal size. The first part is regarded as the first data set, the combination of the first and second parts is viewed as the second data set, the combination of the second data set, and the third part is regarded as the third data set, and so on. The combination of all 10 parts is viewed as the 10th data set. These data sets are used to calculate time consumed by GARA-FS$_H$, GARA-BS$_H$, QGARA-FS$_H$, and QGARA-BS$_H$. The experimental

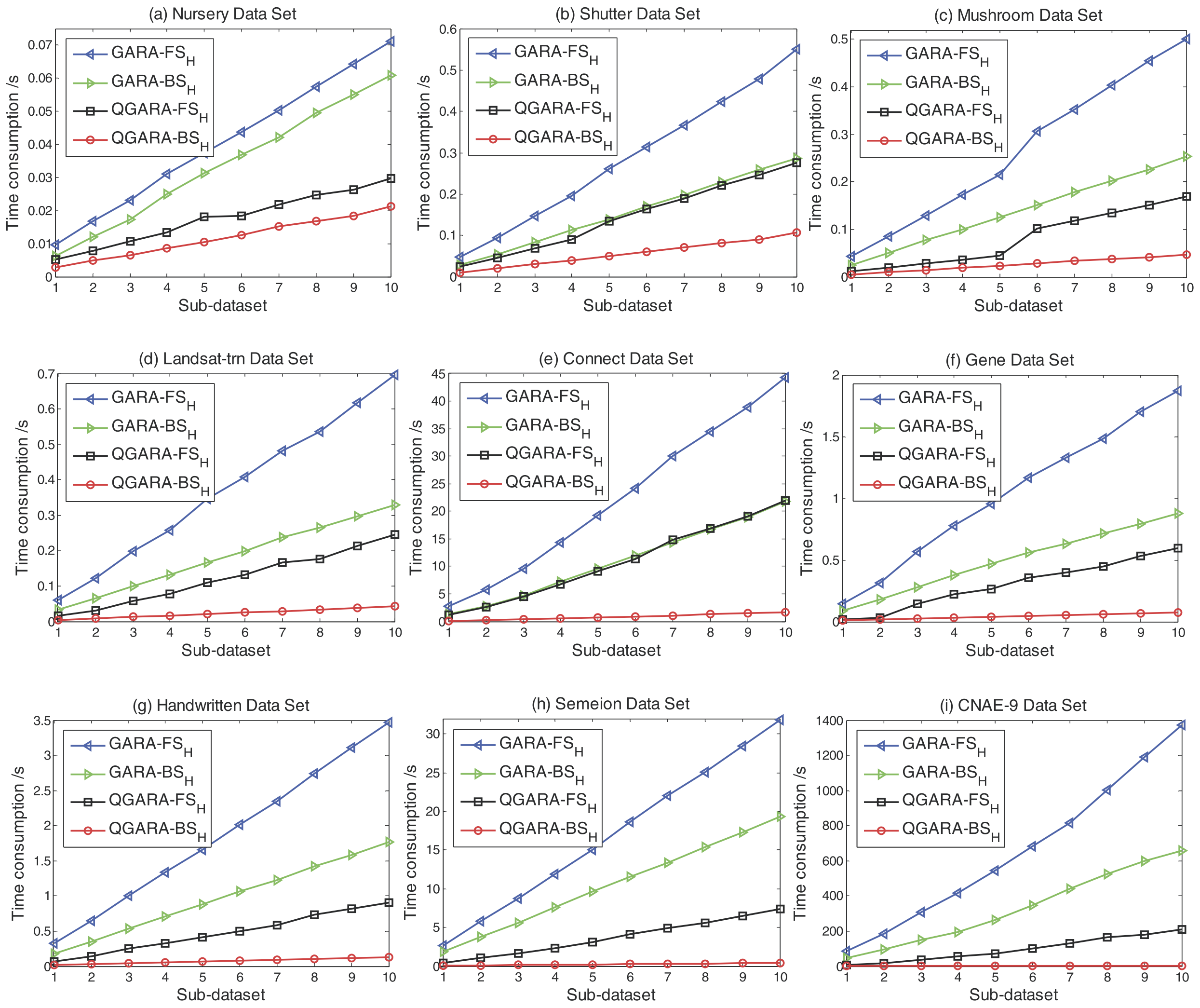

Fig. 2. Comparison of four reduction algorithms for increase in objects.

results of nine data sets are shown in Fig. 2. In all these figures, the horizontal axis represents the size of data set, while the vertical axis refers to the time consumed.

From Fig. 2, it is easy to note that the computing time of each of the four algorithms increases with the increase in the size of data. When dealing with the same proportion data sets, the running time of QGARA-FS$_H$ is less than that of GARA-FS$_H$ and the efficiency of algorithms QGARA-BS$_H$ is higher than that of GARA-BS$_H$. The computational time of QGARA-BS$_H$ is the lowest among the four proposed algorithms.

Moreover, the consuming time of GARA-FS$_H$ and GARA-BS$_H$ increases rapidly with the amount of records increasing, and the increasing time of QGARA-FS$_H$ is relatively low. In addition, it is also easy to see that the slope of the curve of algorithm QGARA-BS$_H$ is very small, such that the increasing time for increasing size of data can be neglected. The four curve shapes in the insets all appear as linear, which is in accordance with the time complexity of these algorithms, that is, $O(|C|^2|U|)$, in which $|C|$ is constant, whereas $|U|$ is variable.

We know that the conditional attributes number of the nine data sets, namely Nursery, Shuttle, Mushroom, Landsat-trn, Connect-4, Gene, Handwritten, Semeion, and CNAE-9 are 8, 9, 22, 37, 42, 60, 64, 256, and 856, respectively. With the increment of the attribute number from Fig. 2(a) to (i), the slopes of the curve for QGARA-BS$_H$ decrease, and the slopes of the data sets Semeion and CNAE-9 are close to zero because of including more conditional attributes.

6.3. Comparison of reduction algorithms for different amount of features

In this subsection, we study the time consumption of the proposed four algorithms with the increase in the number of attributes in terms of computing H-reduct (i.e., R-reduct) on nine data sets, namely Mushroom, Chess, Landsat-trn, Connect-4, Gene, Ticdata2000, Handwritten, Semeion, and CNAE-9. We divide each of these data sets into 10 parts in terms

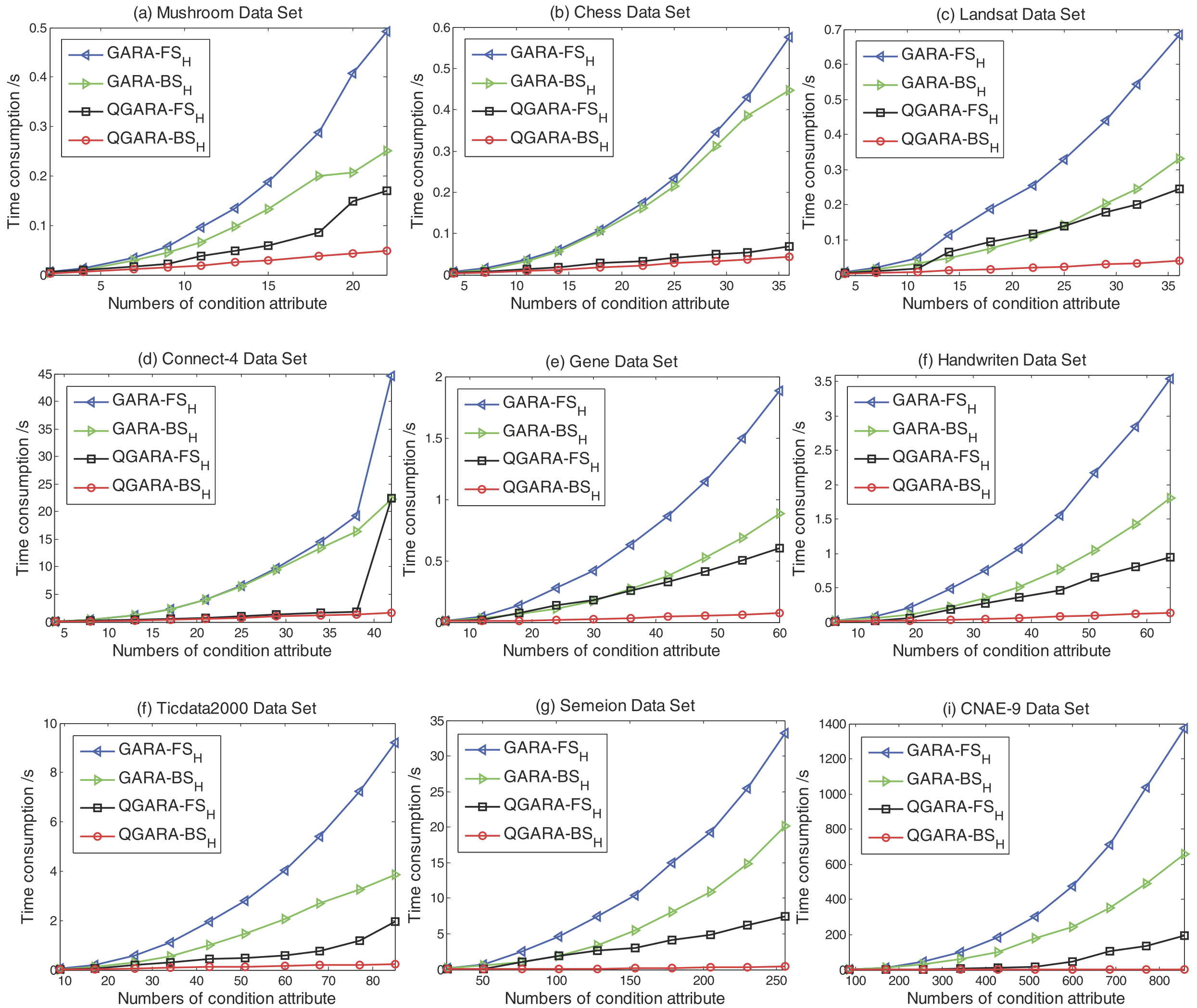

Fig. 3. Comparison of four reduction algorithms when attributes increasing.

of the conditional attributes, and the feature numbers of each part for every data set are shown in Table 10. Hence, the decision table can be divided into 10 data sets as follows.

The size of objects in each data set is the objects number of the original decision table. The combination of the first conditional attribute part and the decision attribute is regarded as the first data set, the combination of the second conditional attribute part and the decision attribute is viewed as the second data set, the combination of the third conditional attribute part and the decision attribute is regarded as the third data set, and so on. The combination of all conditional attributes and the decision attribute is viewed as the 10th data set.

The experimental results of nine data sets are shown in Fig. 3, in which the x-coordinate pertains to the size of data set, while the y-coordinate concerns the time consumed. We can find that the computational time of each of these four algorithms increases with the increase in the number of attributes. However, for the same reduction strategy, the computational time of GARA-FS_H is higher than that of QGARA-FS_H when dealing with the same data set, and the efficiency of algorithms QGARA-BS_H is higher than that of the algorithms of GARA-BS_H when dealing with the same data set.

Moreover, the time consumption of GARA-FS_H and GARA-BS_H increases rapidly with the amount of attributes increasing, and the increasing time of QGARA-FS_H is relatively slow. However, the running time of QGARA-BS_H increases very slowly. The slope of the curve of algorithm QGARA-BS_H is very small, such that the increasing time for increasing amount of attribute can be neglected. In addition, the curve shapes of algorithms GARA-FS_H, GARA-BS_H, and QGARA-FS_H appear conic, which accords with the time complexity of these algorithms, that is, $O(|C|^2|U|)$, in which $|U|$ is constant, whereas $|C|$ is variable.

The conditional attributes number of nine data sets, namely Mushroom, Chess, Landsat-trn, Connect-4, Gene, Ticdata2000, Handwritten, Semeion, and CNAE-9 are 22, 36, 37, 42, 60, 64, 85, 256, and 856, respectively. Similar to Fig. 2, in Fig. 3, with the increment of the attribute number from Fig. 3(a) to (i), the slopes of the curve decrease for QGARA-BS$_H$, and the slopes of data sets Semeion and CNAE-9 are close to zero for including more conditional attributes.

6.4. Comparison of reduction results for five algorithms

In this subsection, we only compare the reduct results of the forward heuristic reduction algorithm and three backward heuristic reduction algorithms. Four attribute reduction algorithms are GARA-FS$_H$, GARA-BS$_H$(s-b), GARA-BS$_H$(b-s), and GARA-BS$_H$(sig), where GARA-FS$_H$ denotes the relative relation reduction method, GARA-BS$_H$(s-b) represents that we define ascending numerical order as the attributes' increasing order of GARA-BS$_H$ algorithm, GARA-BS$_H$(b-s) denotes that we define descending numerical order as the attributes' increasing order of GARA-BS$_H$ algorithm, and GARA-BS$_H$(sig) denotes that we define the attributes' increasing order by $W_H(a|D)$ for every attribute a of GARA-BS$_H$ algorithm. The effects of four reduction algorithms are shown in Table 11 and 12. The integer listed in Table 11 and 12 represents the corresponding attribute position in each specific data set. That is, 1 denotes the first attribute, 2 denotes the second attribute, and so on.

The comparisons show that the reduct results of GARA-FS$_H$ and GARA-BS$_H$ are not always the same in every data set. Moreover, the reduct results of three backward heuristic reduction algorithms (GARA-BS$_H$(s-b), GARA-BS$_H$(b-s) and GARA-BS$_H$(sig)) are within expectation, which also is not necessary to have the same reduct results, because attributes order in three reduction algorithms are different. Furthermore, we can also note the following laws of reduct results. The attribute distribution in the reduct of GARA-BS$_H$(s-b) is more inclined to the latter-part attributes of the conditional attribute set, the attribute distribution in the reduct of GARA-BS$_H$(b-s) prefers the front-part attributes of the conditional attribute set, and the attribute distribution in the reduct of GARA-BS$_H$(sig) distributes more evenly among the conditional attribute set. For example, for Gene data set, the reduct result of GARA-BS$_H$(s-b) is $\{27, 48, 51, 52, 54, 55, 56, 57, 58, 59, 60\}$, the reduct result of GARA-BS$_H$(b-s) is $\{1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 32\}$, and the reduct result of GARA-BS$_H$(sig) is $\{10, 13, 14, 18, 19, 20, 22, 25, 33, 34, 35\}$. Meanwhile, the analogous laws of reduction results are present in the data sets Landsat-trn, Semeion, Shuttle, Handwritten, and CNAE-9.

Measuring significance of the attribute is a key issue for reduct results of any heuristic reduction algorithm. The significance measures are different among these heuristic reduction algorithms. The significance measure of both GARA-FS$_H$ and GARA-BS$_H$(sig) is discernibility degree; the significance measures of FSPA-PR and FSPA-SCE are positive region and conditional entropy, respectively; the significance measures are both the rank of features. For GARA-BSH(s-b) and GARA-BSH(b-s), the attribute significance depends only on before and after order of the attributes. However, the front attribute of features order are not always most important attributes for user requirements. Hence, average lengths of the reduct for algorithms GARA-BSH(s-b) and GARA-BSH(b-s) are both larger than those of other algorithms.

6.5. Comparison of classification accuracies for five algorithms

To evaluate the classification accuracies of different reduction algorithms for different classifiers, we select nine data sets from Table 4 and use the original data set and the corresponding reduced data set to train NB (Naive Bayes), C4.5, and KNN (k-Nearest Neighbor algorithm) classifiers based on 10-fold cross-validation method. The classification accuracies with respect to the raw data and reduced data generated by different algorithms are shown in Tables 13–15, where Raw denotes raw data and the boldface highlights the highest accuracy among different selecting algorithms. The numbers in parentheses rank the five attribute reduction algorithms (including original attribute set) for nine data sets, with the best performing feature subset ranked 1, the second best ranked 2, and so on; in case of a tie, average ranks are assigned.

From Tables 13–15, it is clear that most of the reduced data sets can improve the classification accuracy by eliminating redundant attributes from the original data set. For NB classifier, GARA-FS$_H$ achieves the best classification performance, and the number of the highest classification accuracy is five times out of nine data sets; the number of achieving the most classification accuracy for GARA-BS$_H$(s-b) and GARA-BS$_H$(sig) are both four times among nine data sets in C4.5 and KNN classifiers. In addition, we can find that the best average rank values of classification accuracy for classifiers NB, C4.5, and KNN are 1.94, 2.39, and 2.39, respectively. From average rank of classification accuracy, GARA-FS$_H$ can obtain the best classification performance on NB and C4.5 classifiers; GARA-BS$_H$(sig) can obtain the best classification accuracy on NB classifier.

However, from the results of classification accuracy in Tables 13–15, we could not find the best optimal reduct among different types of attribute reducts, and it is very difficult to determine the reduct obtaining the best classification accuracy on different classifiers. The cause for the above phenomena is that for the same data, different applications may have different needs to meet different properties, which leads to the fact that the reduced results and classification accuracies are possibly different.

7. Conclusions

The reduct is a set of conditional attributes that can keep a certain classification property of the original decision table. To preserve the diversified properties of a decision table, there exist different types of reduct definitions such as relative relation reduct (R-reduct), positive-region reduct (P-reduct), distribution reduct (D-reduct), maximum distribution reduct

Table 11
Result comparisons of four reduct methods.

Data set	Algorithms	# of reduct	Results of reduct
Chess	GARA-FS$_H$	29	1,3,4,5,6,7,9,10,11,12,13,15,16,17,18,20,21,23,24,25,26,27,28,30,31,33,34,35,36
	GARA-BS$_H$(s-b)	29	1,3,4,5,6,7,10,12,13,15,16,17,18,20,21,22,23,24,25,26,27,28,30,31,32,33,34,35,36
	GARA-BS$_H$(b-s)	29	1,3,4,5,6,7,9,10,11,12,13,15,16,17,18,20,21,23,24,25,26,27,28,30,31,33,34,35,36
	GARA-BS$_H$(sig)	29	1,3,4,5,6,7,9,10,11,12,13,15,16,17,18,20,21,23,24,25,26,27,28,30,31,33,34,35,36
Landsat-trn	GARA-FS$_H$	7	1,8,10,12,18,25,35
	GARA-BS$_H$(s-b)	8	26,28,29,31,32,34,35,36
	GARA-BS$_H$(b-s)	9	1,2,3,4,5,6,7,8,11
	GARA-BS$_H$(sig)	8	2,6,10,14,19,20,22,30
Gene	GARA-FS$_H$	10	2,3,6,10,13,17,25,35,45,50
	GARA-BS$_H$(s-b)	11	27,48,51,52,54,55,56,57,58,59,60
	GARA-BS$_H$(b-s)	11	1,2,3,4,5,6,7,9,10,11,32
	GARA-BS$_H$(sig)	11	10,13,14,18,19,20,22,25,33,34,35
Semeion	GARA-FS$_H$	18	7,13,24,27,47,66,104,116,124,136,139,153,158,172,178,220,227,248
	GARA-BS$_H$(s-b)	44	151,159,169,171,173,180,187,197,199,200,204,206,209,212,214,215,216,217,219,220, 225,226,228,231,232,233,234,235,236,238,239,240,242,243,245,246,247,248,249,250,251,253,254,255
	GARA-BS$_H$(b-s)	41	3,5,6,7,8,9,10,11,12,13,14,15,16,17,19,20,21,22,25,26,27,28,29,30,32,33,34,38,40,42, 43,44,48,50,52,57,59,61,63,85,98
	GARA-BS$_H$(sig)	26	6,7,13,14,25,27,30,105,121,130,137,146,158,178,194,210,211,221,227,234,235,236,237,246,247,248
Mushroom	GARA-FS$_H$	4	5,20,21,22
	GARA-BS$_H$(s-b)	4	5,20,21,22
	GARA-BS$_H$(b-s)	4	3,5,7,8,20
	GARA-BS$_H$(sig)	5	3,5,19,20,22
Nursery	GARA-FS$_H$	8	1,2,3,4,5,6,7,8
	GARA-BS$_H$(s-b)	8	1,2,3,4,5,6,7,8
	GARA-BS$_H$(b-s)	8	1,2,3,4,5,6,7,8
	GARA-BS$_H$(sig)	8	1,2,3,4,5,6,7,8
Shuttle	GARA-FS$_H$	4	1,2,3,9
	GARA-BS$_H$(b-s)	4	1,2,3,5
	GARA-BS$_H$(s-b)	4	2,5,8,9
	GARA-BS$_H$(sig)	4	1,2,8,9
Handwritten	GARA-FS$_H$	7	2,14,20,21,46,51,60
	GARA-BS$_H$(s-b)	9	46,51,52,53,54,59,60,61,63
	GARA-BS$_H$(b-s)	9	2,4,5,6,11,12,13,14,18
	GARA-BS$_H$(sig)	8	14,30,38,46,51,52,53,59
Ticdata2000	GARA-FS$_H$	23	1,2,9,18,25,28,31,44,45,47,48,49,54,55,57,58,59,61,63,64,68,80,83
	GARA-BS$_H$(s-b)	24	1,2,28,34,38,40,42,44,47,55,59,66,68,69,70,73,75,78,79,80,82,83,84,85
	GARA-BS$_H$(b-s)	24	1,2,3,6,7,8,14,16,44,45,47,48,49,54,55,57,58,59,61,63,64,68,80,83
	GARA-BS$_H$(sig)	23	1,2,15,18,28,30,39,44,45,47,48,49,54,55,57,59,61,63,64,68,79,80,83
Connect-4	GARA-FS$_H$	36	1,2,3,4,6,7,8,9,10,11,13,14,15,16,17,19,20,21,22,23,24,25,26,27,28,29,31,32,33,34,35,37,38,39,40,42
	GARA-BS$_H$(s-b)	41	2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32, 33,34,35,36,37,38,39,40,41,42
	GARA-BS$_H$(b-s)	34	1,2,3,4,5,7,8,9,10,11,13,14,15,16,17,19,20,21,22,23,25,26,27,28,29,31,32,33,34,35,37,38,39,41
	GARA-BS$_H$(sig)	41	1,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32, 33,34,35,36,37,38,39,40,41,42
CNAE-9	GARA-FS$_H$	76	8,16,21,64,73,74,76,77,78,106,112,120,134,151,154,161,165,192,196,200,202,203,208, 212,259,273,277,322,329,334,335,339,346,348,349,350,351,360,361,372,383,391,394, 404,422,424,444,477,484,488,500,514,519,520,540,547,556,591,608,609,611,616,620, 632,649,651,674,685,706,727,732,769,815,816,824,833
	GARA-BS$_H$(s-b)	95	106,192,203,208,273,277,279,297,329,335,339,343,350,351,353,360,361,366,376,383, 391,404,416,418,422,424,448,470,480,481,484,485,488,512,514,519,520,524,527,532, 542,544,545,547,556,582,591,608,609,611,612,613,615,616,620,632,638,649,651,655, 657,672,674,685,694,702,706,709,724,727,730,732,742,744,745,752,757,758,760,761, 766,768,769,808,810,815,816,822,824,825,833,834,839,840,852
	GARA-BS$_H$(b-s)	105	8,11,16,20,21,25,28,35,40,49,52,60,61,64,69,70,72,74,76,77,78,81,100,102,106,110, 115,120,134,147,151,152,162,164,167,168,175,182,184,192,195,200,202,203,208,212, 231,233,247,253,259,262,269,270,273,277,279,283,299,306,329,333,334,335,338,339, 346,348,349,357,360,361,372,374,376,380,383,388,391,404,408,412,418,422,424,444, 457,463,469,477,484,488,504,519,520,540,547,556,608,611,616,620,674,685,727
	GARA-BS$_H$(sig)	74	8,21,64,74,76,78,103,106,134,151,192,196,200,203,208,212,259,273,277,289,317,322, 329,334,335,339,342,346,349,351,360,361,374,383,391,404,422,424,477,484,491,500, 512,514,519,520,534,540,542,547,556,582,591,596,608,609,611,616,620,632,649,651, 674,706,727,732,769,810,815,816,824,825,833,840

Table 12
Comparison of reduct length for different reduction algorithms.

Data set	GARA-FS$_H$	GARA-BS$_H$(s-b)	GARA-BS$_H$(b-s)	GARA-BS$_H$(sig)	FSPA-PR	FSPA-SCE
Chess	29	29	29	29	29	29
Landsat-trn	7	8	9	8	9	9
Gene	10	11	11	11	10	10
Semeion	18	44	41	26	23	23
Mushroom	4	4	4	5	5	5
Nursery	8	8	8	8	8	8
Shuttle	4	4	4	4	4	4
Handwritten	7	9	9	8	8	10
Ticdata2000	23	24	24	23	23	23
Connect-4	36	41	34	41	35	40
CNAE-9	76	95	105	74	84	86
Average	20.18	25.18	25.27	21.55	21.64	22.45

Table 13
Comparison of classification accuracy by classifier NB.

Data set	Raw	GARA-FS$_H$	GARA-BS$_H$(s-b)	GARA-BS$_H$(b-s)	GARA-BS$_H$(sig)
Chess	87.891%(5)	**90.300%(1.5)**	88.204%(4)	88.2673%(3)	**90.300%(1.5)**
Landsat-trn	67.215%(5)	**82.594%(1)**	76.550%(3)	78.038%(4)	81.352%(2)
Gene	75.455%(3)	65.078%(4)	**90.878%(1)**	62.884%(5)	85.297%(2)
Semeion	**75.248%(1)**	69.178%(2)	43.942%(5)	52.354%(4)	59.636%(3)
Mushroom	76.342%(4)	**88.854%(1)**	86.854%(3)	73.215%(5)	88.220%(2)
Shuttle	99.527%(4)	**99.727%(1)**	99.609%(2)	86.353%(5)	99.557%(3)
Handwritten	81.335%(2)	62.689%(3)	55.392%(4)	50.908%(5)	83.096%(1)
Ticdata2000	78.718%(5)	89.849%(3)	89.248%(4)	90.141%(2)	**90.261%(1)**
Connect-4	71.142%(4)	**72.312%(1)**	71.073%(3)	72.172%(2)	71.927%(5)
Average rank	3.89	**1.94**	3.22	3.39	2.39

Table 14
Comparison of classification accuracy by classifier C4.5.

Data set	Raw	GARA-FS$_H$	GARA-BS$_H$(s-b)	GARA-BS$_H$(b-s)	GARA-BS$_H$(sig)
Chess	99.037%(3)	**99.093%(2.5)**	**99.093%(2.5)**	**99.093%(2.5)**	**99.093%(2.5)**
Landsat-trn	72.266%(5)	**78.979%(1)**	75.242%(4)	75.919%(3)	78.444%(2)
Gene	62.884%(5)	90.357%(2)	**94.282**%(1)	63.324%(4)	73.793%(3)
Semeion	**75.832%(1)**	67.922%(2)	50.219%(5)	57.627%(4)	58.945%(3)
Mushroom	**100.000%(3)**	**100.000%(3)**	**100.000%(3)**	**100.000%(3)**	**100.000%(3)**
Shuttle	99.604%(5)	99.640%(3)	99.650%(2)	**99.726%(1)**	99.616%(4)
Handwritten	**80.694%(1)**	65.320%(3)	64.324%(4)	57.794%(5)	72.259%(2)
Ticdata2000	93.885%(4)	93.961%(2)	**93.971%(1)**	93.954%(3)	93.954%(5)
Connect-4	80.970%(4)	81.028%(3)	80.631%(5)	81.026%(2)	**81.063%(1)**
Average rank	3.44	**2.39**	3.06	3.06	2.83

Table 15
Comparison of classification accuracy by classifier KNN.

Data set	Raw	GARA-FS$_H$	GARA-BS$_H$(s-b)	GARA-BS$_H$(b-s)	GARA-BS$_H$(sig)
Chess	96.871%(5)	**97.403%(1.5)**	97.309%(3.5)	97.309%(3.5)	**97.403%(1.5)**
Landsat-trn	72.289%(5)	75.558%(3)	73.145%(4)	**85.164%(1)**	79.436%(2)
Gene	61.944%(5)	74.546%(2)	67.868%(3)	62.257%(4)	**84.483%(1)**
Semeion	**71.463%(1)**	70.182%(2)	53.107%(5)	58.569%(4)	63.842%(3)
Mushroom	**100.000%(3)**	**100.000%(3)**	**100.000%(3)**	**100.000%(3)**	**100.000%(3)**
Shuttle	99.681%(5)	99.736%(3)	99.700%(4)	99.760%(2)	**99.783%(1)**
Handwritten	**88.612%(1)**	63.932%(4)	64.395%(3)	57.722%(5)	74.911%(2)
Ticdata2000	89.728%(4)	**90.038%(1)**	90.021%(2)	89.986%(3)	89.608%(5)
Connect-4	79.353%(3)	77.622%(4)	73.554%(5)	**80.921%(1)**	79.364%(2)
Average rank	3.56	2.61	3.61	2.94	**2.39**

(Md-reduct), and assignment reduct (A-reduct). Many scholars have presented various quick reduction algorithms to attain different reducts. However, there is lack of a quick general reduction algorithm structure to derive different reducts.

In this study, we focus on researching the generalized quick reduction algorithms for five representative attribute reducts from the viewpoint of the relative discernibility. First, a general definition of the decision table for five representative reducts (R-reduct, P-reduct, D-reduct, Md-reduct, and A-reduct) is presented to construct unified definition of various reducts easily. On the basis of the common definition of the decision table, the concept of relative discernibility and the corresponding computing approach for the relative discernibility by the indiscernibility are introduced. Then, we propose the notion of the general relative discernibility reduct for the general decision table and prove the equivalence between the relative discernibility reduct and the discernibility matrix reduct. Two general attribute reduction strategies based on the relative discernibility are presented to obtain the corresponding reducts. One is the forward addition reduction algorithm (GARA-FS$_\triangle$), which is a down–top search strategy and the other is the backward elimination reduction algorithm (GARA-BS$_\triangle$), which is a top–down search strategy. However, GARA-FS$_\triangle$ and GARA-BS$_\triangle$ are still undesirable for the huge-volume and high-dimensional data sets. To further increase efficiency of reduction algorithms, subsequently, we introduce an accelerator for reduction algorithms, which mainly reduces the number of radix sort for improving equivalence partitioning. On the basis of the accelerator, we design two quick general reduction algorithms based on the relative discernibility: quick general forward heuristics attribute reduction algorithm (QGARA-FS$_\triangle$) and backward heuristics attribute reduction algorithm (QGARA-BS$_\triangle$). A series of comparative experiments with UCI data sets are conducted to evaluate the performance of the proposed reduction algorithms. The numerical experimental results show that the general reduction framework can satisfy various types of reducts and two reduction algorithms embedded that the new accelerator can efficiently reduce the time consumed by attribute reduction.

It is noted that the proposed reduction algorithms of this study are only suitable for complete inconsistent decision tables. However, in real world, many decision tables are both inconsistent and incomplete. It is necessary to develop reduction theory and approach for inconsistent and incomplete decision tables. Therefore, in the future work, we will study and develop quick general reduction framework in the inconsistent incomplete decision table for obtaining R-reduct, P-reduct, D-reduct, Md-reduct, A-reduct, and so on.

Acknowledgements

This study was partially supported by grants from the National Science Foundation of China (Nos. 5130711 and 61402005), funds from Anhui Provincial Natural Science Foundation (Nos. 1308085QF114, 1508085MF126, and 1508085-MF127), funds from Anhui Education Natural Science Foundation (No. KJ2013A015), funds from Excellent Young Foundation of ChuZhou University (No. 2013RC003), and by the fund from Initial Scientific Research of Chuzhou University for Ge Hao.

References

[1] R.B. Bhatt, M. Gopal, On fuzzy-rough sets approach to feature selection, Pattern Recognit. Lett. 26 (7) (2005) 965–975.
[2] Y.M. Chen, D.Q. Miao, R.Z. Wang, A rough set approach to feature selection based on power set tree, Knowl.-Based Syst. 24 (3) (2011) 275–281.
[3] M. Dash, H. Liu, Consistency-based search in feature selection, Artif. Intell. 151 (1–2) (2003) 155–176.
[4] U.M. Fayyad, L.P. Shapiro, P. Smyth, R. Uthurusamy, Advances in Knowledge Discovery and Data Mining, AAAI Press/the MIT Press, 1996.
[5] H. Ge, L.S. Li, C.J. Yang, A efficient attribute reduction algorithm based on conflict region, Chinese J. Comput. 35 (2) (2012) 342–350.
[6] X.H. Hu, N. Cercone, Learning in relational databases a rough set approach, Int. J. Comput. Intell. 11 (2) (1995) 323–338.
[7] Y.L. Jiang, Z.X. Yang, Y. Liu, Quick distribution reduction algorithm in inconsistent information system, Acta Autom. Sin. 38 (3) (2012) 382–388.
[8] M. Kryszkiewicz, Comparative studies of alternative type of knowledge reduction in inconsistent systems, Int. J. Intell. Syst. 16 (1) (2001) 105–120.
[9] M. Kryszkiewicz, Certain, generalized decision, and membership distribution reducts versus functional dependencies in incomplete systems, in: Proceedings of Rough Sets and Intelligent Systems Paradigms, 2007, pp. 162–174.
[10] F. Li, Y.Q. Yin, Approaches to knowledge reduction of covering decision systems based on information theory, Inf. Sci. 179 (11) (2009) 1694–1704.
[11] M. Li, C.X. Shang, S.Z. Feng, J.P. Fan, Quick attribute reduction in inconsistent decision tables, Inf. Sci. 254 (2014) 155–180.
[12] J.Y. Liang, J.R. Mi, W. Wei, F. Wang, An accelerator for attribute reduction based on perspective of objects and attributes, Knowl.-Based Syst. 44 (2013) 90–100.
[13] J.Y. Liang, Z.B. Xu, The algorithm on knowledge reduction in incomplete information systems, Int. J. Uncertain. Fuzziness Knowl.-Based Syst. 10 (1) (2002) 95–103.
[14] J.Y. Liang, Z.Z. Shi, D.Y. Li, M.J. Wierman, The information entropy, rough entropy and knowledge granulation in incomplete information systems, Int. J. Gen. Syst. 35 (6) (2006) 641–654.
[15] H. Liu, H. Motoda, Feature Selection for Knowledge Discovery and Data Mining, Kluwer Academic Publishers, 1998.
[16] Q.H. Liu, F. Li, F. Min, M. Ye, G.W. Yang, Efficient knowledge reduction algorithm based on new conditional information entropy, Control Decis. 20 (8) (2005) 878–882.
[17] Q.H. Liu, L.T. Chen, J.Y. Zhang, F. Min, Knowledge reduction in inconsistent decision tables, in: The International Conference on Advanced Data Mining and Applications, ADMA 2006, Xi'an, China, in: LNAI, vol. 4093, 2006, pp. 626–635.
[18] S.H. Liu, Q.J. Seng, B. Wu, Z.Z. Shi, F. Hu, Research on efficient algorithms for Rough set methods, Chinese J. Comput. 26 (5) (2003) 524–529.
[19] Y. Liu, R. Xiong, J. Chu, Quick attribute reduction algorithm with hash, Chinese J. Comput. 32 (8) (2009) 1494–1499.
[20] D.Q. Miao, Y. Zhao, Y.Y. Yao, H.X. Li, F.F. Xu, Relative reducts in consistent and inconsistent decision tables of the Pawlak rough set model, Inf. Sci. 179 (2009) 4140–4150.
[21] D.Q. Miao, G.R. Hu, A heuristic algorithm for reduction of knowledge, J. Comput. Res. Dev. 36 (6) (1999) 681–684.
[22] Z.Q. Meng, Z.Z. Shi, A fast approach to attribute reduction in incomplete decision systems with tolerance relation-based rough sets, Inf. Sci. 179 (16) (2009) 2774–2793.
[23] Z.Q. Meng, Z.Z. Shi, Extended rough set-based attribute reduction in inconsistent incomplete decision systems, Inf. Sci. 204 (10) (2012) 44–69.
[24] Z. Pawlak, Rough sets, Int. J. Comput. Inf. Sci. 11 (1982) 341–356.
[25] Z. Pawlak, Rough Sets Theoretical Aspects of Reasoning About Data, Kluwer Academic Publishers, 1991.
[26] J. Qian, D.Q. Miao, Z.H. Zhang, W. Li, Hybrid approaches to attribute reduction based on indiscernibility and discernibility relation, Int. J. Approx. Reason. 52 (2) (2011) 212–230.
[27] Y.H. Qian, J.Y. Liang, W. Pedrycz, C.Y. Dang, Positive approximation an accelerator for attribute reduction in rough set theory, Artif. Intell. 174 (2010) 597–618.
[28] Y.H. Qian, J.Y. Liang, W. Pedrycz, C.Y. Dang, An efficient accelerator for attribute reduction from incomplete data in rough set framework, Pattern Recognit. 44 (8) (2011) 1658–1670.
[29] R. Quinlan, Induction of decision rules, Mach. Learn. 1 (1) (1986) 81–106.
[30] Q. Shen, A. Chouchoulas, A modular approach to generating fuzzy rules with reduced attributes for the monitoring of complex systems, Eng. Appl. Artif. Intell. 13 (3) (2000) 263–278.

[31] A. Skowron, C. Rauszer, The discernibility matrices and functions in information systems, intelligent decision support, in: Handbook of Applications and Advances of the Rough Sets Theory, Kluwer Academic Publishers, 1992.

[32] G.J. Song, G. Li, An algorithm of attribute reduct based on rough set, Appl. Mech. Mater. (2013) 347–350.

[33] R.W. Swiniarski, A. Skowron, Rough set methods in feature selection and recognition, Pattern Recognit. Lett. 24 (2003) 833–849.

[34] G.Y. Wang, Rough Set Theory and Knowledge Acquisition, Xi'an Jiaotong University Press, 2001.

[35] L. Wei, J.J. Qi, Relation between concept lattice reduction and rough set reduction, Knowl.-Based Syst. 23 (8) (2010) 934–938.

[36] Z.Y. Xu, Z.P. Liu, B.Y. Yang, W. Song, A quick attribute reduction algorithm with complexity of $\max(O(|C||U|), O(|C|^2|U/C|))$, Chinese J. Comput. 29 (3) (2006) 611–615.

[37] M. Yang, An incremental updating algorithm for attribute reduction based on improved discernibility matrix, Chinese J. Comput. 30 (5) (2007) 815–822.

[38] C.S. Yang, L. Shu, Attribute reduction algorithm of incomplete decision table based on tolerance relation, Comput. Technol. Dev. 16 (9) (2006) 68–72.

[39] Y.Y. Yao, Y. Zhao, J. Wang, On reduct construction algorithms, in: Rough Sets and Knowledge Technology, in: Lecture Notes in Computer Science, 2006, pp. 297–304.

[40] D.Y. Ye, Z.J. Chen, A new type of attribute reduction for inconsistent decision tables and its computation, Int. J. Uncertain. Fuzziness Knowl.-Based Syst. 18 (2) (2010) 209–222.

[41] W.X. Zhang, J.S. Mi, W.Z. Wu, Knowledge reduction in inconsistent information systems, Chinese J. Comput. 26 (2003) 12–18.

[42] W.X. Zhang, J.S. Mi, W.Z. Wu, Approaches to knowledge reductions in inconsistent systems, Int. J. Intell. Syst. 18 (2003) 989–1000.

[43] Y. Zhao, Y.Y. Yao, Data analysis based on discernibility and indiscernibility, Inf. Sci. 177 (2007) 4959–4976.

[44] J.W. Zhou, B.Q. Feng, Y. Liu, New algorithm for quick computing core, J. Xi'an Jiaotong Univ. 41 (6) (2007) 688–691.

[45] J. Zhou, D.Q. Miao, W. Pedrycz, H.Y. Zhang, Analysis of alternative objective functions for attribute reduction in complete decision tables, Soft Comput. 15 (2011) 1601–1616.