Soft Comput (2011) 15:1601–1616
DOI 10.1007/s00500-011-0690-7

ORIGINAL PAPER

Analysis of alternative objective functions for attribute reduction in complete decision tables

Jie Zhou · Duoqian Miao · Witold Pedrycz · Hongyun Zhang

Published online: 10 February 2011

Abstract Attribute reduction and reducts are important notions in rough set theory that can preserve discriminatory properties to the highest possible extent similar to the entire set of attributes. In this paper, the relationships among 13 types of alternative objective functions for attribute reduction are systematically analyzed in complete decision tables. For inconsistent and consistent decision tables, it is demonstrated that there are only six and two intrinsically different objective functions for attribute reduction, respectively. Some algorithms have been put forward for minimal attribute reduction according to different objective functions. Through a counterexample, it is shown that heuristic methods cannot always guarantee to produce a minimal reduct. Based on the general definition of discernibility function, a complete algorithm for finding a minimal reduct is proposed. Since it only depends on reasoning mechanisms, it can be applied under any objective function for attribute reduction as long as the corresponding discernibility matrix has been well established.

J. Zhou (✉) · D. Miao · H. Zhang
Department of Computer Science and Technology,
Tongji University, Shanghai 201804,
People's Republic of China
e-mail: jie_jpu@163.com

D. Miao
e-mail: miaoduoqian@163.com

H. Zhang
e-mail: zhanghongyun583@sina.com

J. Zhou · W. Pedrycz
Department of Electrical and Computer Engineering,
University of Alberta, Edmonton, AB T6G 2G7, Canada
e-mail: pedrycz@ee.ualberta.ca

Keywords Complete decision table · Objective function for attribute reduction · Discernibility function · Minimal reduct

1 Introduction

Rough set theory (Pawlak 1982; Pawlak and Skowron 2007) has developed nearly three decades and has had widespread success in many research areas, such as pattern recognition, machine learning, knowledge acquisition, economic forecast, data mining, etc. (An et al. 1996; Beynon and Peel 2001; Dimitras et al. 1999; Jelonek et al. 1995). It aims at data analysis problems involving uncertain or imprecise information and becomes one of major schemes of granular computing (Pedrycz 2007; Pedrycz et al. 2008). Attribute reduct is one of the most fundamental and important notions in rough set theory, which can preserve a certain property of an original decision table in the same way as entire condition attribute set.

Generally, decision tables can be positioned into two categories: complete decision tables and incomplete decision tables (Zakowski 1993). A decision table is complete if each object does not have any default values over all attributes, otherwise, it is incomplete. Complete decision tables can be further divided into two categories: consistent and inconsistent decision tables (Pawlak et al. 1988). A decision table is consistent if all object pairs that have the same condition values also have the same decision value, otherwise, it is inconsistent.

Diversified properties embedded in a decision table can be revealed from different profiles. A sort of formulation description cannot represent all the properties. Many objective functions for attribute reduction have been proposed by means of some special facets in a complete decision table (Kryszkiewicz 2001; Li and Zhang 2004; Miao and Wang 1997; Miao et al. 2009; Pawlak and Skowron 2007; Slezak 2000; Wang et al. 2002, 2005; Yao and Zhao 2008; Zhang et al. 2003). Classical objective function for attribute reduction proposed by Pawlak (1982) focuses on the remaining positive region or the quality of classification. Miao and Wang (1997) put forward a new objective function from the view of information entropy. Wang et al. (2002) further presented conditional information entropy for describing attribute reduction. Slezak (2000) constructed some approaches for attribute reduction based on attribute frequencies in decision tables, in which distribution reduct was introduced. Zhang et al. (2003) proposed the notion of maximal distribution reduct and possible reduct. Therein, possible reduct preserves the upper approximation of each decision class. Some types of knowledge reduction for a single object and an entire decision table were, respectively, compared by Kryszkiewicz (2001). For the latter, approximate reduct which preserves the decision values for each object and possible reduct which distinguishes each object from objects that do not belong to the relevant upper approximation are both presented. Recently, the notion of relative relationship preservation reduct was provided by Miao et al. (2009).

Each reduct definition introduced above only describes a special profile of a decision table. It is necessary to clarify their relationships since some of them are essentially equivalent. Kryszkiewicz (2001) firstly investigated some alternative objective functions for attribute reduction. Unfortunately, some results are not reasonable. Based on Kryszkiewicz's research works, the related results were theoretically improved by Li and Zhang (2004). The relationship among distribution reduct, maximal distribution reduct and possible reduct was discussed by Zhang et al. (2003). Wang (2003) studied the differences between algebraic view and information view for attribute reduction. The equivalence between distribution reduct and condition information entropy reduct was proved by Qin et al. (2005). Miao et al. (2009) revealed the relationship among three distinct relative reduct definitions in consistent and inconsistent decision tables. The relationship among absolute attribute reduct and some relative attribute reducts was discussed by Deng et al. (2007). Wang et al. (2008) presented a systematic study on attribute reduction based on general binary relations in rough set theory. These researches often focus on the relationship among some chosen alternative objective functions for attribute reduction. The comprehensive relationship among available typical objective functions for attribute reduction in complete decision tables still needs more investigation.

More than one reduct often exists in a decision table under a given attribute reduction objective function. Final decision rule sets are derived relying on the obtained reducts directly. The conciseness, understandability, generality and precision of the decision rule sets will be distinct according to different reducts, so some optimal results are expected, i.e., the minimal reducts which have the shortest length. Such that redundant attributes can be removed as much as possible, the storage space for the decision table can be managed effectively and the properties of the decision rule set will become excellent. Unfortunately, searching for a minimal reduct has been proved to be an NP-hard problem (Wong and Ziarko 1985).

Many heuristics have been proposed and investigated for finding an optimal reduct or approximate optimal reduct (Hu and Cercone 1995; Nguyen and Nguyen 1996; Skowron and Rauszer 1991; Thangavel and Pethalakshmi 2009; Wang and Wang 2001; Xu et al. 2006; Yao and Zhao 2009; Yao et al. 2008; Ye and Chen 2006; Zhao and Yao 2007). However, Wang and Miao (1998) has illustrated that heuristic algorithms are incomplete to find a minimal reduct; in other words, a minimal reduct is not always attained by heuristic algorithms when a decision table is given. Sometimes, the result is just a superset of a reduct (Yao et al. 2006). Possibly, in order to construct a minimal reduct, the set of all reducts can be obtained at the first step. In this case, the time and space complexity will be very high.

In this paper, the available 13 typical forms of objective functions for attribute reduction in complete decision tables are systematically investigated. It is illustrated that there are only six and two intrinsically different objective functions for attribute reduction in inconsistent and consistent decision tables, respectively. According to the general reduct definition (Zhao et al. 2007) and the general definition of discernibility matrix (Miao et al. 2009), a complete algorithm *CAMARDF* for minimal attribute reduction is presented. The efficiency of the proposed algorithm is elucidated by experiments involving both UCI (Asuncion and Newman 2007) and synthetic data sets.

The paper is organized as follows. Some preliminaries are reviewed in Sect. 2. Section 3 presents 13 typical forms of attribute reduct definitions. The relationship among them in consistent and inconsistent decision tables are investigated, respectively. In Sect. 4, the failure mechanisms of heuristic algorithms for finding a minimal reduct are analyzed in detail. In Sect. 5, a complete algorithm for minimal attribute reduction is proposed based on the discernibility matrix. Some experimental results are presented in Sect. 6, while main conclusions are delivered in Sect. 7.

2 Preliminaries

For convenience, some basic concepts in rough set theory are briefly recalled in this section.

Definition 1 Decision table DT can be represented as the tuple: $DT = (U, C \cup D, V, \rho)$, where $U = \{x_1, x_2, \ldots, x_n\}$ is a finite nonempty set of objects, called the universe; C and D are the finite nonempty sets of condition and decision attributes, respectively, $C \cap D = \emptyset$; $V = \cup_{a \in (C \cup D)} V_a$ is the union of attribute value domains, V_a is a nonempty set of values for attribute a; $\rho : U \times (C \cup D) \rightarrow V$ is an information function, $\rho(x, a)$ denotes the value of object x on attribute a.

$\forall B \subseteq C$ denotes $IND(B) = \{(x, y) | \forall b \in B, \rho(x, b) = \rho(y, b)\}$, then $IND(B)$ is a equivalence relation on U, referred to as an indiscernibility relation. $U/IND(B) = \{[x]_B | x \in U\}$ is the collection of all equivalence classes w.r.t. (with respect to) B, denoted as U/B briefly. $[x]_B = \{y | y \in U, (x, y) \in IND(B)\}$ is the equivalence class determined by x w.r.t. B. Each element of U/B and U/D is called as condition class w.r.t. B and decision class, respectively. Obviously, the partition U/C has the finest information granules and the partition $U/\emptyset$ has the coarsest information granules.

Especially, we assume that $D = \{d\}$ with only one decision attribute. If a decision table has more than one decision attribute, it is easy to convert it to an equivalent decision table with one decision attribute which values are determined by the combination of the original decision values.

Definition 2 Given a decision table $DT = (U, C \cup D, V, \rho)$, for $x_i \in U$ $(i = 1, 2, \ldots, n)$, $\forall x_j \in U (j \neq i)$, if $(x_i, x_j) \in IND(C)$, then it has $\rho(x_i, d) = \rho(x_j, d)$, x_i is called a consistent object w.r.t. C, otherwise, x_i is called an inconsistent object w.r.t. C.

Definition 3 Given a decision table $DT = (U, C \cup D, V, \rho)$ and $\forall B \subseteq C$, the partition of universe U w.r.t. B is denoted as $U/B = \{E_1, E_2, \ldots, E_{|U/B|}\}$. For each condition class $E_i \in U/B$ $(i = 1, 2, \ldots, |U/B|)$, if $\forall x \in E_i$, x is a consistent object w.r.t. B, then E_i is called a consistent condition class w.r.t. B, otherwise, E_i is called an inconsistent condition class w.r.t. B. The notation $|X|$ denotes the cardinality of set X.

Definition 4 Given a decision table $DT = (U, C \cup D, V, \rho)$, $\forall x \in U$, if x is a consistent object w.r.t. C, then DT is called a consistent decision table, otherwise, DT is called an inconsistent decision table.

For an inconsistent decision table, its objects can be divided into two groups. One is composed of consistent objects and the other is composed of inconsistent objects. In this case, consistent decision tables can be considered as a special case of inconsistent decision tables. Object, equivalence class and decision table can be established as a hierarchy with three levels from lower to higher, or from concrete to abstract.

Definition 5 Given a decision table $DT = (U, C \cup D, V, \rho)$, $\forall X \subseteq U$ and $\forall B \subseteq C$, the upper and lower approximations of set X w.r.t. B are denoted as $\overline{B}(X)$, $\underline{B}(X)$, respectively, and defined as:

$$\begin{aligned} \overline{B}(X) &= \cup\{E_i | E_i \cap X \neq \emptyset, E_i \in U/B\}, \\ \underline{B}(X) &= \cup\{E_i | E_i \subseteq X, E_i \in U/B\}. \end{aligned} \tag{1}$$

The upper approximation of X is composed of objects that belong to set X possibly, and the lower approximation of X is composed of objects that belong to set X certainly. The upper and the lower approximations of X approximate the concept X from two sides. In other words, the concept X can be approximately described by two sets. Especially, if the concept X is uncertain or vague, such approximate description has important meaning.

3 Attribute reduction in complete decision tables

In what follows, we elaborate on 13 typical kinds of objective functions for attribute reduction in complete decision tables. Some main results about attribute reduction in incomplete decision tables can be found in Kryszkiewicz (1998), Liang et al. (2006, 2008), Leung and Li (2003), and Qian et al. (2009).

3.1 Alternative objective functions for attribute reduction

An attribute reduct is a minimal subset of entire condition attribute set that are jointly sufficient and individually necessary for preserving a certain property of a given decision table (Zhao et al. 2007). Formally, a general definition of an attribute reduct can be described as follows.

Definition 6 Given a decision table $DT = (U, C \cup D, V, \rho)$ and a certain property Δ of DT under consideration. The attribute subset $A \subseteq C$ is called a reduct of C, if it satisfies the following conditions:

1. evaluation function e for Δ is $2^{C \cup D} \rightarrow L$, which maps an attribute set to an element of a poset L;
2. $e(A) = e(C)$;
3. $\forall A' \subset A$, $e(A') \neq e(A)$.

Condition (2) indicates the joint sufficiency of attribute set A, namely, attribute set A is sufficient to preserve the property Δ. Condition (3) means each element in A is individually necessary as remaining the property. If the certain property Δ of DT under consideration is regarding to the decision attribute set D, A is called a relative reduct of C, otherwise, it is called a absolute reduct of C.

The property Δ can be interpreted from diverse profiles of a decision table. Different objective functions for attribute reduction can be constructed according to different properties. No matter what properties will be considered, conditions (2) and (3) must be satisfied at the same time.

(I) *Absolute reduct*. Given a decision table $DT = (U, C \cup D, V, \rho)$, $A \subseteq C$ is a absolute reduct (Komorowski et al. 1999) of C, iff A satisfies the following conditions:

1. $IND(A) = IND(C)$;
2. for any attribute subset $A' \subset A$, $IND(A') \neq IND(A)$.

The collection of equivalence classes U/C will remain the same according to the definition of absolute reduct. This definition is often used in information systems which have no decision attributes. In fact, the jointly sufficient condition for absolute reduct is most strict. Any condition class in U/C cannot be changed during the whole process of attribute reduction.

(II) *Relationship preservation reduct*. $\forall A \subseteq C$, the relative indiscernibility relation defined by A w.r.t. D, is described as:

$$IND(A|D) = \{(x, y) \in U \times U | (\forall a \in A \rightarrow \rho(x, a) = \rho(y, a)) \vee \rho(x, d) = \rho(y, d)\}. \quad (2)$$

Obviously, a relative indiscernibility relation is not an equivalence relation since the transitive property is not satisfied.

Given a decision table $DT = (U, C \cup D, V, \rho)$, $A \subseteq C$ is a relationship preservation reduct (Miao et al. 2009) of C, iff A satisfies the following conditions:

1. $IND(A|D) = IND(C|D)$;
2. for any attribute subset $A' \subset A$, $IND(A'|D) \neq IND(A|D)$.

Object pairs in $IND(A|D)$ will share the same condition values or the same decision value. If an object pair belongs to $IND(A)$, then it must belongs to $IND(A|D)$. Contrarily, it may be not satisfied. Compared with absolute reduct restrictions, the jointly sufficient condition for relationship preservation reduct is looser.

(III) *Positive region reduct*. $\forall A \subseteq C$, the positive region of D w.r.t. A, is defined as:

$$POS_A(D) = \bigcup_{D_j \in U/D} \underline{A}(D_j), \quad (3)$$

where $U/D = \{D_1, D_2, \ldots, D_{|U/D|}\}$. Given a decision table $DT = (U, C \cup D, V, \rho)$, $A \subseteq C$ is a positive region reduct (Komorowski et al. 1999) of C, iff A satisfies the following conditions:

1. $POS_A(D) = POS_C(D)$;
2. for any attribute subset $A' \subset A$, $POS_{A'}(D) \neq POS_A(D)$.

Positive region reduct is the classical attribute reduct definition proposed by Pawlak. Positive region is composed of all consistent objects in a decision table. For a consistent decision table $POS_C(D) = U$, it indicates that all objects in a consistent decision table can be classified definitely w.r.t. decision attribute set. For an inconsistent decision table $POS_C(D) \subset U$, in this case, some objects cannot be classified with certainty w.r.t. decision attributes.

(IV) *Classification quality reduct*. $\forall A \subseteq C$, the quality of classification of DT w.r.t. A, is defined as:

$$\gamma_A(D) = \frac{|POS_A(D)|}{|U|}. \quad (4)$$

Given a decision table $DT = (U, C \cup D, V, \rho)$, $A \subseteq C$ is a classification quality reduct (Komorowski et al. 1999) of C, iff A satisfies the following conditions:

1. $\gamma_A(D) = \gamma_C(D)$;
2. for any attribute subset $A' \subset A$, $\gamma_{A'}(D) \neq \gamma_A(D)$.

$\gamma_C(D)$ is also called the degree of dependency of attribute set D w.r.t. attribute set C. The quality of classification γ is a quantitative description for the ability of classification of decision tables. Virtually, it measures the ratio of objects that can be classified certainly w.r.t. D in the universe. Obviously, $0 \leq \gamma_C(D) \leq 1$. For a consistent decision table, its quality of classification is equal to 1.

(V) *Condition entropy reduct*. $\forall A \subseteq C$, the condition information entropy of DT w.r.t. A, is defined as:

$$H(D|A) = -\sum_{i=1}^{|U/A|} P(X_i) \sum_{j=1}^{|U/D|} P(D_j|X_i) \log(P(D_j|X_i)), \quad (5)$$

where $X_i \in U/A$ $(i = 1, 2, \ldots, |U/A|)$, $D_j \in U/D$ $(j = 1, 2, \ldots, |U/D|)$, $P(X_i) = \frac{|X_i|}{|U|}$, $P(D_j|X_i) = \frac{|D_j \cap X_i|}{|X_i|}$.

Given a decision table $DT = (U, C \cup D, V, \rho)$, $A \subseteq C$ is a condition entropy reduct (Wang et al. 2002, 2005) of C, iff A satisfies the following conditions:

1. $H(D|A) = H(D|C)$;
2. for any attribute subset $A' \subset A$, $H(D|A') \neq H(D|A)$.

The uncertainty of a decision table is predominantly caused by the conflict objects, viz., inconsistent objects. The holistic uncertainty of a decision table can be depicted by condition information entropy $H(D|C)$. Obviously, $0 \leq H(D|C) \leq \log(n)$, where n is the number of objects. For a consistent decision table, all objects can be certainly classified with respect to decision attribute set, so $|D_j \cap C_i|/|C_i| = 1$ or 0 for all $C_i \in U/C$ and $D_j \in U/D$. Then it has $H(D|C) = 0$, namely, the holistic uncertainty of a consistent decision table is equal to zero. For an inconsistent decision table, $0 < H(D|C) \leq \log n$. $H(D|C)$ will achieve the maximal value if the whole universe is partitioned into one equivalence class under condition attribute set C and all objects in universe have totally different decision values at the same time.

(VI) *Probability distribution reduct*. $\forall x \in U$, the membership distribution function of object x over all decision classes w.r.t. $A \subseteq C$, is defined as:

$$\mu_A(x) = (P(D_1|[x]_A), P(D_2|[x]_A), \ldots, P(D_{|U/D|}|[x]_A)), \quad (6)$$

where $D_j \in U/D$ $(j = 1, 2, \ldots, |U/D|)$, $P(D_j|[x]_A) = \frac{|D_j \cap [x]_A|}{|[x]_A|}$.

Given a decision table $DT = (U, C \cup D, V, \rho)$, $A \subseteq C$ is a probability distribution reduct (Slezak 2000) of C, iff A satisfies the following conditions:

1. $\forall x \in U$, it has $\mu_A(x) = \mu_C(x)$;
2. for any attribute subset $A' \subset A$, $\exists x' \in U$, such that $\mu_{A'}(x') \neq \mu_A(x')$.

Obviously, $\mu_A(x)$ can be considered as the probability distribution on U/D. Probability distribution reduct preserves the degree to which each object belongs to each decision class.

(VII) *Maximal distribution reduct*. $\forall x \in U$, the maximal distribution decision function of object x w.r.t. $A \subseteq C$, is defined as:

$$\phi_A(x) = \left\{ D_j | \frac{[x]_A \cap D_j}{[x]_A} = \max_{k=1}^{|U/D|} \left\{ \frac{[x]_A \cap D_k}{[x]_A} \right\} \right\}. \quad (7)$$

Given a decision table $DT = (U, C \cup D, V, \rho)$, $A \subseteq C$ is a maximal distribution reduct (Zhang et al. 2003) of C, iff A satisfies the following conditions:

1. $\forall x \in U$, it has $\phi_A(x) = \phi_C(x)$;
2. for any attribute subset $A' \subset A$, $\exists x' \in U$, such that $\phi_{A'}(x') \neq \phi_A(x')$.

The value of $\max_{k=1}^{|U/D|} \left\{ \frac{[x]_A \cap D_k}{[x]_A} \right\}$ can be considered as the degree of confidence of uncertain rules derived from the equivalence class $[x]_A$. A maximal distribution reduct preserves all decision rules with maximal confidence degree derived from each equivalence class in U/C.

(VIII) *Decision value preservation reduct*. $\forall x \in U$, its generalized decision value w.r.t. $A \subseteq C$, is defined as:

$$\delta_A(x) = \{\rho(y, d) | y \in [x]_A\}. \quad (8)$$

Given a decision table $DT = (U, C \cup D, V, \rho)$, $A \subseteq C$ is a decision value preservation reduct (Miao et al. 2009) of C, iff A satisfies the following conditions:

1. $\forall x \in U$, it has $\delta_A(x) = \delta_C(x)$;
2. for any attribute subset $A' \subset A$, $\exists x' \in U$, such that $\delta_{A'}(x') \neq \delta_A(x')$.

A decision value preservation reduct is also called an approximate reduct in Kryszkiewicz (2001). It preserves the generalized decision value of each object and distinguishes each object from objects that have different generalized decision value.

(IX) *Lower approximation distribution reduct*. The concepts of β lower and upper distribution reducts based on variable precision rough sets were first introduced by Mi et al. (2004). Similarly, the lower approximation distribution of $U/D = \{D_1, D_2, \ldots, D_{|U/D|}\}$ w.r.t. attribute set $A \subseteq C$ in the classical rough set model is denoted as:

$$\underline{\psi}_A(D) = \left(\underline{A}(D_1), \underline{A}(D_2), \ldots, \underline{A}(D_{|U/D|})\right). \quad (9)$$

Given a decision table $DT = (U, C \cup D, V, \rho)$, $A \subseteq C$ is a lower approximation distribution reduct of C, iff A satisfies the following conditions:

1. $\underline{\psi}_A(D) = \underline{\psi}_C(D)$;
2. for any attribute subset $A' \subset A$, $\underline{\psi}_{A'}(D) \neq \underline{\psi}_A(D)$.

The deterministic rules in a decision table can be derived from the lower approximation of each decision class. In other words, a lower approximation distribution reduct will preserve all deterministic rules for a decision table.

(X) *Upper approximation distribution reduct*. The upper approximation distribution of $U/D = \{D_1, D_2, \ldots, D_{|U/D|}\}$ w.r.t. attribute set $A \subseteq C$ is denoted as:

$$\bar{\psi}_A(D) = \left(\bar{A}(D_1), \bar{A}(D_2), \ldots, \bar{A}(D_{|U/D|})\right). \quad (10)$$

Given a decision table $DT = (U, C \cup D, V, \rho)$, $A \subseteq C$ is a upper approximation distribution reduct of C, iff A satisfies the following conditions:

1. $\bar{\psi}_A(D) = \bar{\psi}_C(D)$;
2. for any attribute subset $A' \subset A$, $\bar{\psi}_{A'}(D) \neq \bar{\psi}_A(D)$.

The upper approximation of each decision class will decide associated deterministic rules and some probabilistic rules of a decision table, in this case, the deterministic and probabilistic rules will both be preserved.

(XI) *Sum of lower approximation reduct*. The sum of lower approximation over all decision classes w.r.t. attribute set $A \subseteq C$ is denoted as:

$$\underline{\omega}_A(D) = \frac{1}{|U|} \sum_{j=1}^{|U/D|} \left|\underline{A}(D_j)\right|. \quad (11)$$

Given a decision table $DT = (U, C \cup D, V, \rho)$, $A \subseteq C$ is a sum of lower approximation reduct of C, iff A satisfies the following conditions:

1. $\underline{\omega}_A(D) = \underline{\omega}_C(D)$;
2. for any attribute subset $A' \subset A$, $\underline{\omega}_{A'}(D) \neq \underline{\omega}_A(D)$.

Obviously, $0 \leq \underline{\omega}_A(D) \leq 1$. If all objects in the decision table are inconsistent under attribute set A, then $\underline{\omega}_A(D) = 0$. In this case, no deterministic decision rules will be derived. For consistent decision tables, $\underline{\omega}_A(D) = 1$.

Since $\underline{A}(D_i) \cap \underline{A}(D_j) = \emptyset \quad (i \neq j)$, $|POS_A(D)| = \left|\cup_{D_j \in U/D} \underline{A}(D_j)\right| = \sum_{D_j \in U/D} \left|\underline{A}(D_j)\right|$, so it has $\underline{\omega}_A(D) = \gamma_A(D)$.

(XII) *Sum of upper approximation reduct*. The sum of upper approximation over all decision classes w.r.t. attribute set $A \subseteq C$ is denoted as:

$$\bar{\omega}_A(D) = \frac{1}{|U|} \sum_{i=1}^{|U/D|} \left|\bar{A}(D_j)\right|. \tag{12}$$

Given a decision table $DT = (U, C \cup D, V, \rho)$, $A \subseteq C$ is a sum of upper approximation reduct of C, iff A satisfies the following conditions:

1. $\bar{\omega}_A(D) = \bar{\omega}_C(D)$;
2. for any attribute subset $A' \subset A$, $\bar{\omega}_{A'}(D) \neq \bar{\omega}_A(D)$.

It can be found that $1 \leq \bar{\omega}_A(D) \leq |U/D|$. $\bar{\omega}_A(D)$ will attain the maximal value when the generalized decision value of each object includes all decision values. A sum of upper approximation reduct is also called possible reduct in Zhang et al. (2003) which is different from the following definition formally.

(XIII) *Possible reduct*. Given a decision table $DT = (U, C \cup D, V, \rho)$, $A \subseteq C$ is a possible reduct (Kryszkiewicz 2001) of C, iff A satisfies the following conditions:

1. $\forall x \in U$, it has $[x]_A \subseteq \bar{C}(D_k)$, where $x \in D_k$, $D_k \in U/D$;
2. for any attribute subset $A' \subset A$, $\exists x' \in U$, such that $[x']_{A'} \not\subseteq \bar{C}(D'_k)$, where $x' \in D'_k$, $D'_k \in U/D$.

A possible reduct distinguishes each object $x \in U$ from other objects that not belong to the upper approximation of the decision class including x.

Some other alternative objective functions for attribute reduction in complete decision tables, such as μ-decision reduct, μ-reduct (Kryszkiewicz 2001) can found the equivalent descriptions from introduced 13 types of reduct definitions. As stressed, non-parameter-based objective functions for attribute reduction introduced above will not be compared with parameter-based objective functions, such as α-reduct, α-relative reduct (Nguyen and Slezak 1999), pan-generalized decision reduct (Li and Zhang 2004), etc. In some special cases, a parameter-based objective function can be converted to the corresponding non-parameter-based objective function.

3.2 The relationships among alternative 13 types of attribute reduction in inconsistent decision tables

Some comparative researches on objective functions for attribute reduction in complete decision tables have been done. However, the relationships only among some chosen objective functions from introduced 13 types of attribute reduction are presented in the available researches. In what follows, the comprehensive relationships among introduced 13 types of attribute reduction in inconsistent decision tables will be revealed.

Before moving into details, the overview of the relationships among available 13 kinds of attribute reduction in inconsistent decision tables is given in Fig. 1. The notation "*property*1 $\rightarrow$ *property*2" denotes that if *property*1 is satisfied, then *property*2 will be satisfied as well.

According to the available research results, the main findings can be described as follows:

③ $\leftrightarrow$ ④ was proved by Qin et al. (2005) and Xu et al. (5);

④ $\rightarrow$ ⑤, ④ $\rightarrow$ ⑥, ⑥ $\leftrightarrow$ ⑧ were proved by Zhang et al. (2003);

⑥ $\leftrightarrow$ ⑨ was proved by Li and Zhang (2004);

⑥ $\rightarrow$ ⑩ was proved by Miao et al. (2009).

⑩ $\leftrightarrow$ ⑪ and ⑪ $\leftrightarrow$ ⑬ are obviously satisfied according to the formulas (3), (4) and (11). So only ① $\rightarrow$ ②, ② $\rightarrow$ ③, ⑦ $\leftrightarrow$ ⑧ and ⑩ $\leftrightarrow$ ⑫ need to be proved.

(1) ① $\rightarrow$ ②

Proof According to the definitions of indiscernibility relation and relative indiscernibility relation, this relationship is satisfied. □

However, ② $\rightarrow$ ① is not satisfied. It can be illustrated by the following counterexample.

From decision table I (Table 1), $C = \{a_1, a_2\}$ and $D = \{d\}$. It can be found that $IND(\{a_2\}|D) = IND(\{a_1, a_2\}|D)$. Since $(x_1, x_2) \in IND(\{a_2\})$ and $(x_1, x_2) \notin IND(\{a_1, a_2\})$, so $IND(\{a_2\}) \neq IND(\{a_1, a_2\})$.

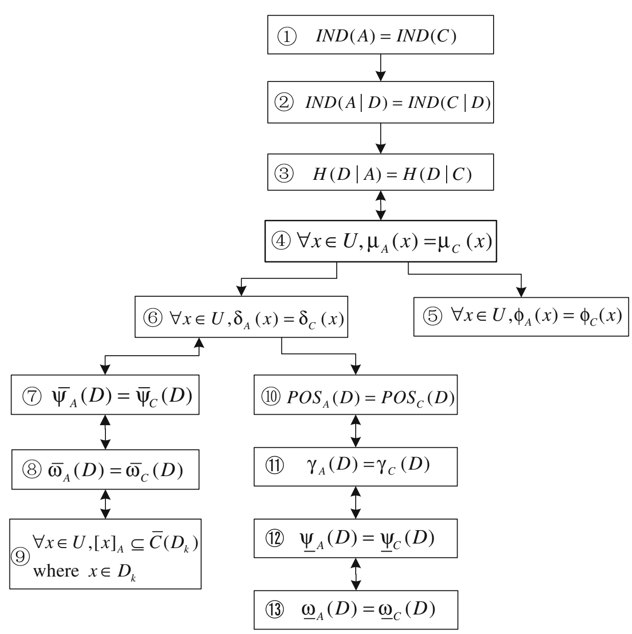

Fig. 1 The relationships among 13 types of attribute reduction in inconsistent decision tables

Table 1 Decision table I

U	a_1	a_2	d
x_1	1	1	0
x_2	2	1	0
x_3	0	0	1

Table 2 Decision table II

U	a_1	a_2	d
x_1	0	0	0
x_2	0	0	1
x_3	0	1	0
x_4	0	1	1

(2) ② → ③

Proof Suppose $C_i, C_j \in U/C \quad (i \neq j)$ will be merged under attribute set $A \subset C$. There are three cases:

(a) C_i and C_j are consistent condition classes. It comes with two cases:
If the decision value of C_i and C_j are the same, it has $IND(A|D) = IND(C|D)$ obviously;
If the decision value of C_i and C_j are different, $\forall x \in C_i$ and $\forall y \in C_j$, it has $(x, y) \notin IND(C|D)$. After merging C_i and C_j under A, $(x, y) \in IND(A|D)$. So in this case, $IND(A|D) \neq IND(C|D)$.

(b) C_i is consistent condition class and C_j is inconsistent condition class.
$\forall x \in C_i$, $\exists y \in C_j$, such that $\rho(x, d) \neq \rho(y, d)$, so $(x, y) \notin IND(C|D)$. After merging C_i and C_j under A, $\forall a \in A$, it has $\rho(x, a) = \rho(y, a)$, so $(x, y) \in IND(A|D)$. Consequently, $IND(A|D) \neq IND(C|D)$.
The case when C_i is inconsistent condition class and C_j is consistent condition class can be proved in a similar fashion.

(c) C_i and C_j are inconsistent condition classes.
$\exists x \in C_i$, $\exists y \in C_j$, such that $\rho(x, d) \neq \rho(y, d)$, so $(x, y) \notin IND(C|D)$. After merging C_i and C_j under A, $\forall a \in A$, it has $\rho(x, a) = \rho(y, a)$, so $(x, y) \in IND(A|D)$. Consequently, $IND(A|D) \neq IND(C|D)$.

□

From the above discussions, only under the case that C_i and C_j are consistent condition classes with the same decision value, it will have $IND(A|D) = IND(C|D)$. Wang et al. (2002) and Wang (2003) have illustrated that $H(D|A) = H(D|C)$ iff C_i and C_j are consistent condition classes with the same decision value or C_i and C_j are inconsistent condition classes but their membership distribution over all decision classes are the same. Thus, it can be concluded that if $IND(A|D) = IND(C|D)$ is satisfied, then $H(D|A) = H(D|C)$ will be satisfied. □

Conversely, if $H(D|A) = H(D|C)$, $IND(A|D) = IND(C|D)$ is not always satisfied. From decision table II (Table 2), $C = \{a_1, a_2\}$ and $D = \{d\}$. It can be checked that $H(D|\{a_1\}) = H(D|\{a_1, a_2\})$. However, $(x_2, x_3) \notin IND(\{a_1, a_2\}|D)$, but $(x_2, x_3) \in IND(\{a_1\}|D)$. Namely, $IND(\{a_1\}|D) \neq IND(\{a_1, a_2\}|D)$.

(3) ⑦ ↔ ⑧

Proof ⑦ → ⑧

Since $\forall D_j \in U/D$, it has $\bar{A}D_j = \bar{C}D_j$, so $|\bar{A}D_j| = |\bar{C}D_j|$. Consequently, $\frac{1}{|U|}\sum_{j=1}^{|U/D|} |\bar{A}D_j| = \frac{1}{|U|}\sum_{j=1}^{|U/D|} |\bar{C}D_j|$, namely, $\bar{\omega}_A(D) = \bar{\omega}_C(D)$.

⑦ ← ⑧

$\forall D_j \in U/D$, it has $\bar{A}D_j \supseteq \bar{C}D_j$. Since $\sum_{j=1}^{|U/D|} |\bar{A}D_j| = \sum_{j=1}^{|U/D|} |\bar{C}D_j|$, so $\forall D_j \in U/D$, it has $\bar{A}D_j = \bar{C}D_j$, namely, $\bar{\psi}_A(D) = \bar{\psi}_C(D)$. $$

(4) ⑩ ↔ ⑫

Proof ⑩ → ⑫

Since $\cup_{D_j \in U/D} \underline{A}D_j = \cup_{D_j \in U/D} \underline{C}D_j$ and $\forall D_j \in U/D$, it has $\underline{A}D_j \subseteq \underline{C}D_j$, so $\forall D_j \in U/D$, $\underline{A}D_j = \underline{C}D_j$. Namely, $\underline{\psi}_A(D) = \underline{\psi}_C(D)$.

⑩ ← ⑫ can be directly derived according to formula (3) and (9). □

All implication and equivalence relationships in Fig. 1 have been proven. Essentially, some objective functions for attribute reduction are equivalent according to Fig. 1. Thirteen types of objective functions can be grouped to only six categories that is {①,②,{③,④},⑤,{⑥,⑦,⑧,⑨}, {⑩,⑪,⑫,⑬}}

During the process of attribute reduction, some condition classes w.r.t. C will be merged. The mergence will be diversified under different objective functions.

Under objective function ①, all the condition classes w.r.t. C will remain the same during the whole process of attribute reduction. Each of them cannot be merged with others no matter whether it is inconsistent or consistent condition class.

Under objective function ②, only consistent condition classes w.r.t. C with the same decision value can be merged during the process of attribute reduction. However, any inconsistent condition class under C must remain the same.

Under objective functions {③,④}, consistent condition classes w.r.t. C with the same decision value or inconsistent condition classes w.r.t. C which have the same membership distribution over all decision classes can be merged during the process of attribute reduction.

Under objective function ⑤, all condition classes, no matter whether they are consistent or inconsistent, if they have the same maximal distribution decision function over all decision classes, then they can be merged during the process of attribute reduction. It means that a consistent condition class and an inconsistent condition class may be merged after removing some attributes.

Under objective functions {⑥,⑦,⑧,⑨}, consistent condition classes w.r.t. C with the same decision value or inconstant condition classes w.r.t. C which have the same generalized decision value, no matter what their membership distribution over all decision classes, can be merged during the process of attribute reduction.

Under objective functions {⑩,⑪,⑫,⑬}, consistent condition classes w.r.t. C with the same decision value can be merged and inconsistent condition classes w.r.t. C can be randomly merged no matter what their membership distribution over all decision classes and their generalized decision value.

From the top to the bottom in Fig. 1, the restrictions for attribute reduction become looser. Only under objective function ⑤, a consistent condition class and an inconsistent condition class may be across merged after reducing some attributes. Under any other objective functions, the mergence between a consistent condition class and an inconsistent condition class is illegal.

3.3 The relationships among alternative 13 types of attribute reduction in consistent decision tables

Consistent decision tables can be considered as a special kind of inconsistent decision tables. The overview of relationships among 13 types of attribute reduction in consistent decision tables can be adjusted as illustrated in Fig. 2.

Some available results can be described as follows:

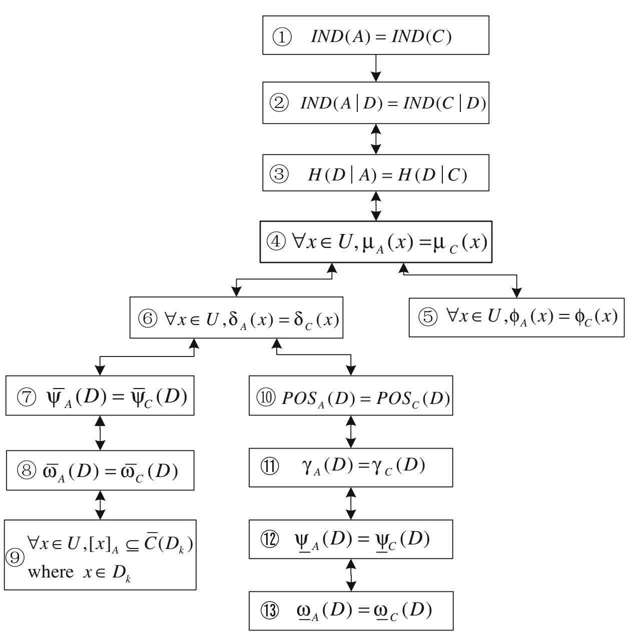

Fig. 2 The relationships among 13 types of attribute reduction in consistent decision tables

⑩ ↔ ⑥ and ⑥ ↔ ② were proved by Miao et al. (2009); ⑩ ↔ ③ was proved by Wang (2003) and Wang et al. (2005).

④ ↔ ⑤ is instantaneously satisfied.

Thus, Fig. 2 can be directly derived from Fig. 1. From Fig. 2, there are only two intrinsically different types of objective functions for attribute reduction in consistent decision tables {①,{②,③,④,⑤,⑥,⑦,⑧,⑨,⑩,⑪,⑫,⑬}}. They can be briefly considered as absolute reduct and relative reduct, respectively. The situations of mergence between condition classes during the process of attribute reduction in consistent decision tables are simper than in inconsistent decision tables. For absolute reduct, all the condition classes w.r.t. C will remain the same during the whole process of attribute reduction which is as the same as in inconsistent decision tables. For relative reduct, only the condition classes w.r.t. C with the same decision value can be merged during the subsequent process of attribute reduction.

4 Completeness of heuristic algorithms for finding minimal reducts

In intelligent computing, heuristic algorithms are often applied to search for optimal or approximate optimal results for NP-hard problems. The next node that is considered as the most hopeful for last optimal result will be chosen to expand. Heuristics play an important role in the entire procedures since it decides which node will be chosen to extend. On the one hand, it affects the efficiency of problem solving. On the other hand, it impacts the quality of result, namely, the search result should be close to the optimum as much as possible.

Generally, reducts embedded in a decision table are not unique. Many heuristic attribute reduction algorithms have been put forward in order to get an optimal one, namely a minimal reduct. No matter which attribute reduction objective functions will be applied, a general heuristic algorithm for attribute reduction with addition strategy (Zhao et al. 2007) can be outlined as follows:

Algorithm 1: general heuristic algorithm model for attribute reduction

Input: decision table $DT = (U, C \cup D, V, \rho)$;

Output: reduct RED preserving the property Δ.

Step1: $RED = Core$, $C = C - RED$;

Step2: if $C = \emptyset$ or termination conditions w.r.t. Δ are satisfied, goto Step6;

Step3: compute the significance for each attribute in C under property Δ;

Step4: select an attribute a from C which has the maximal attribute significance;

Step5: $RED = RED \cup \{a\}$, $C = C - \{a\}$, goto Step2;

Step6: output RED.

Step 3 plays an important role in the algorithm because it decides which attribute will be extended, namely, it decides upon the search path. If the extended attribute is not suitable, the search path will deviate from the optimal one.

The quantification of attribute significance in Step 3 is realized according to the property Δ. In order to illustrate the completeness of heuristic algorithms for minimal attribute reduction, the classical reduct definition that preserves the positive region in a consistent decision table will be discussed here. Other reduct definitions in consistent or inconsistent decision table can be illustrated in a similar way.

Given a decision table $DT = (U, C \cup D, V, \rho)$ as shown in Table 3, where $D = \{d\}$ is the decision attribute, $C = \{a_1, a_2, \ldots, a_{13}\}$ is the set of condition attributes.

$\forall B \subseteq C$, the significance of attribute $a \in (C - B)$ w.r.t. B, is defined as:

$$SIG(a, B, D) = \gamma(B \cup \{a\}, D) - \gamma(B, D). \tag{13}$$

If $B = \emptyset$, then $\gamma(B, D) = 0$. It can be checked that the core set $Core = \{a_{11}\}$. After running algorithm 1 for attribute reduction, the result comes in the form $\{a_{11}, a_1, a_3, a_2, a_4\}$. However, the minimal reduct of Table 3 is $\{a_{11}, a_2, a_3, a_4\}$ and this demonstrates the heuristic algorithm fails to obtain a minimal reduct.

After a careful analysis of the general heuristic reduction model, the attribute with the maximal significance is selected to extend preferentially based on core set (in Table 3, the attribute a_1 will be chosen to extend preferentially after core attribute $\{a_{11}\}$ being computed). Thus, the search path may deviate from the optimal one. At the preliminary stage of search process, choosing the attribute with maximal significance to extend can reduce the search space fast and local optimum can be guaranteed. However, the global optimization cannot be assured, namely, the search path will not always be optimal.

Table 3 Decision table III

U	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}	a_{12}	a_{13}	d
x_1	0	0	0	0	0	0	0	0	0	0	0	0	0	Y
x_2	1	0	1	0	0	0	0	0	0	0	0	0	0	N
x_3	1	1	0	1	0	0	0	0	0	0	0	0	0	N
x_4	1	0	0	1	0	0	0	1	0	0	0	0	0	N
x_5	1	1	0	0	0	0	0	0	1	0	0	0	0	N
x_6	0	0	1	0	0	0	1	0	0	0	0	1	0	N
x_7	0	0	1	0	0	1	0	0	0	0	0	0	1	N
x_8	0	1	0	0	1	0	0	0	0	0	0	0	0	N
x_9	0	0	0	1	0	0	0	0	0	1	0	0	0	N
x_{10}	0	0	0	0	0	0	0	0	0	0	1	0	0	N

Essentially, heuristic algorithms are a special group of greedy algorithms. In order to find global optimization, the local optimum must be gradually modified. In the process of searching a minimal reduct, after the search path beginning from the attribute with maximal significance being considered, the search paths beginning from the attribute with the second largest significance, the third largest significance, etc., should also be considered one by one.

5 Minimal attribute reduction based on discernibility function

Skowron and Rauszer (1991) have proved that the reducts of a decision table are in one-to-one correspondence with the prime implicants of corresponding discernibility function. For consistent decision tables, Skowron has constructed discernibility matrices for both absolute reduct and relative reduct, respectively. For inconsistent decision tables, the discernibility matrices for objective functions ① and ⑩ were proposed by Skowron; Zhang et al. (2003) investigated the discernibility matrices for objective functions ④, ⑤ and ⑥; Miao et al. (2009) studied the discernibility matrix for objective function ②. So under each objective function for attribute reduction, two types in consistent decision tables and six types in inconsistent decision tables, the associated discernibility matrix can be constructed. According to them, a general definition of discernibility matrix was provided by Miao et al. (2009).

Definition 7 Given a decision table $DT = (U, C \cup D, V, \rho)$ and a certain property Δ of DT. The discernibility matrix $\mathbf{M}_\Delta = (\mathbf{M}_\Delta(x, y))$ w.r.t. Δ is a $|U| \times |U|$ matrix, in which the element $\mathbf{M}_\Delta(x, y)$ for an object pair (x, y) satisfies:

$$\mathbf{M}_\Delta(x,y) = \begin{cases} \{a \in C \mid \rho(x,a) \neq \rho(y,a)\} & (x,y) \text{are distinguishable w.r.t.} \Delta \\ \emptyset & \text{otherwise} \end{cases} \tag{14}$$

$\mathbf{M}_\Delta$ is a symmetric matrix. We can only use its lower or upper triangle values to describe it. After establishing the discernibility matrix, the corresponding discernibility function can be directly obtained by disjunction and conjunction operations.

Definition 8 Given a decision table $DT = (U, C \cup D, V, \rho)$ and a certain property Δ of DT, $\mathbf{M}_\Delta$ is the discernibility matrix w.r.t. Δ, the corresponding discernibility function of DT is a Boolean function defined as follows:

$$DF(DT) = \wedge\{\vee c_{ij} : 1 \leq i < j \leq n, c_{ij} \neq \emptyset\}, \tag{15}$$

where c_{ij} is an element in $\mathbf{M}_\Delta$. $\vee c_{ij} = \vee a(a \in c_{ij})$ is the disjunction of all attribute variables $a \in c_{ij}$. Absorption law is often applied to simplify the discernibility function. If $(a \vee b) \wedge (a \vee b \vee c)$ is included in discernibility function DF, then the clause $a \vee b \vee c$ will be removed. The simplified discernibility function is also a conjunctive normal form. The problem of finding minimal reducts is polynomially equivalent to the problem of searching prime implicants with the shortest length in discernibility function. A prime implicant of a Boolean function is an implicant that cannot be covered by a more general implicant.

Some attribute reduction algorithms were proposed based on discernibility matrices. Chang et al. (1999) presented an attribute reduction approach based on a discernibility matrix and logic computation to get the best attribute reducts that satisfy user's demand. However, how to transform a conjunction norm form to a disjunction norm form is not introduced. Nguyen and Nguyen (1996) proposed some efficient approximate algorithms for minimal reduct problem which include Johnson's strategy and random strategy. Since these approximate algorithms often generate a superreduct, some irrelevant attributes in the obtained result must be eliminated. Wang and Miao (1998) also pointed out that the approximate algorithms (heuristic algorithms) are incomplete for minimal attribute reduction problem. As being discussed before, more information on attributes in the discernibility function should be considered in order to obtain a proper minimal reduct completely and effectively.

In the sequel, the simplified discernibility function will be used to discuss. Suppose a simplified discernibility function $DF = f_1 \wedge f_2 \wedge \cdots \wedge f_s$, we consider $DF = \{f_1, f_2, \ldots, f_s\}$ instead and if $f_i = a_1 \vee a_2 \vee \cdots \vee a_{k_i}$, we consider $f_i = \{a_1, a_2, \ldots, a_{k_i}\}$ instead when no confusion can arise. The set of all variables in DF is denoted as Ω_{DF}.

Theorem 1 *Given a decision table $DT = (U, C \cup D, V, \rho)$ and a certain property Δ of DT, its associated discernibility function $DF = f_1 \wedge f_2 \wedge \cdots \wedge f_s$. $A \subseteq C$ is a reduct of DT w.r.t. Δ, then $\forall f_i \in DF\,(i = 1, 2, \ldots, s)$, $A \cap f_i \neq \emptyset$.*

Theorem 1 follows from Skowron's notions directly. That is to say, a reduct of a decision table must have some common items with each clause of discernibility function.

Theorem 2 *Given a decision table $DT = (U, C \cup D, V, \rho)$ and a certain property Δ of DT, its associated discernibility function $DF = f_1 \wedge f_2 \wedge \cdots \wedge f_s$. $\forall a \in \Omega_{DF}$, the set of the shortest implicants that include attribute a is denoted as $I(a)$, the set of minimal reducts of DT is denoted as $MR(DT)$, then $MR(DT)$ satisfies:*

$$MR(DT) = \cup\,\{I(a) | a \in \Omega_{DF},\ for\ \forall b \in \Omega_{DF} - \{a\},\ \forall \xi \in I(a), \forall \xi' \in I(b), such\ that |\xi| \leq |\xi'|\}. \quad (16)$$

Since prime implicants are special cases of implicants, the implicants with the shortest length are also the prime implicants with the shortest length in a Boolean function. Apparently, Theorem 2 is satisfied. The right side of (16) indicates that the length of each element in $I(a)$ is the shortest compared with other variables in DF. Thus the problem of searching for shortest prime implicants can be transformed to searching for shortest implicants. There is no need to restrict search space only to prime implicants.

Why we concentrate on the implicants, not the prime implicants? The reason is that the elements in $I(a)$ are not always prime implicants. For example, a discernibility function is given as follows:

$$DF = (a_1 \vee a_3) \wedge (a_1 \vee a_2 \vee a_4) \wedge (a_1 \vee a_2 \vee a_8) \wedge (a_3 \vee a_7) \wedge (a_3 \vee a_6) \wedge (a_2 \vee a_5),$$

where $a_1 \wedge a_2 \wedge a_3 \in I(a_1)$, but it is not a prime implicant of DF (due to the prime implicant $a_2 \wedge a_3$), so it is not a reduct under consideration. According to Theorem 2, it is no need to cost much to check whether the elements in $I(a)$ are prime implicants or not.

A complex problem is often solved through some simple sub-problems and the mechanism "divide and conquer" can be applied. According to distributive law and associative law in logic reasoning, a Boolean function can be divided as follows.

Theorem 3 (expansion law) (Starzyk et al. 2000) *Given a decision table $DT = (U, C \cup D, V, \rho)$ and a certain property Δ of DT, its associated discernibility function $DF = f_1 \wedge f_2 \wedge \cdots \wedge f_s$, $\forall a \in \Omega_{DF}$, DF can be decomposed w.r.t. a as follows:*

$$DF = DF_1 \vee DF_2, \quad (17)$$

where

$$DF_1 = \wedge\{f | f \in DF \wedge a \notin f\} \wedge a, \quad (18)$$

$$DF_2 = \wedge\{f | f \in DF \wedge a \notin f\} \wedge \{(f - \{a\}) | f \in DF \wedge a \in f\}. \quad (19)$$

According to the expansion law, the implicants of discernibility function DF can be divided into two groups in light of each variable $a \in \Omega_{DF}$. One includes variable a, can be derived from DF_1. The others do not include variable a, can be derived from DF_2. The shortest implicants including variable a can only be derived from DF_1, namely, it just need to find the shortest implicants of $\wedge\{f | f \in DF \wedge a \notin f\}$ in DF_1. Thus, Theorems 2 and 3 can be applied repeatedly.

This iterative process is called as *decomposition principles* of the discernibility function and the variable a is called a *decomposition variable*. If all variables in Ω_{DF} are considered, the shortest implicants will be obtained.

Theorem 4 *Suppose two conjunctive normal forms* $F = f_1 \wedge f_2 \wedge \cdots \wedge f_s$ *and* $F' = F \wedge f$, f *is a new clause that is composed of the disjunction of some variables.* ξ *and* ξ' *are the shortest prime implicant of* F *and* F', *respectively*, *then* $|\xi| \le |\xi'|$.

Proof The variables in the new clause f only have three cases as follows:

1. $\forall a \in f$, $a \notin \Omega_F$, then $\xi' = \xi \wedge a$, $|\xi'| = |\xi| + 1 > |\xi|$;
2. if $\exists a \in f$ and $a \in \xi$, then $\xi \cap f \ne \emptyset$, it has $\xi' = \xi$, $|\xi'| = |\xi|$;
3. if $\forall a \in \{b | b \in f \wedge b \in \Omega_F\} \ne \emptyset$ and $a \notin \xi$, the shortest implicants that include attribute a in F are denoted as $I_F(a)$, then $\forall \xi_a \in I_F(a)$, it has $|\xi_a| \ge |\xi|$. If $|\xi_a| = |\xi|$, then the case is as the same as (2); if $|\xi_a| = |\xi| + 1$, then we can have $\xi' = \xi_a$, $|\xi'| > |\xi|$; if $|\xi_a| > |\xi| + 1$, then we can have $\xi' = \xi \wedge a$, $|\xi'| = |\xi| + 1$, $|\xi'| > |\xi|$.

From the above discussions, it can be concluded that $|\xi'| \ge |\xi|$.

Theorem 4 indicates that the length of the shortest prime implicants of conjunctive normal form will increase monotonically as the number of clauses increased. In other words, it implies that the length of the minimal reducts will increase monotonically as the number of the clauses of discernibility function increased.

Theorem 5 *Given a decision table* $DT = (U, C \cup D, V, \rho)$ *and a certain property* Δ *of* DT, *its associated discernibility function is* $DF = f_1 \wedge f_2 \wedge \ldots \wedge f_s$. $\forall a \in \Omega_{DF}$, *if* $P_{DF}(a) = 1$, *then*:

1. *if* $\exists f \in DF$, *such that* $a \in f$ *and* $|f| = 1$, *then* $I(a) = MR(DT)$;
2. *if* $\exists f \in DF$, *such that* $a \in f$ *and* $|f| > 1$. *If* $\forall b \in (f - \{a\})$, $P_{DF}(b) = 1$, *then* $\forall \xi_a \in I(a)$, $\forall \xi_b \in I(b)$ *and* $\forall \xi \in MR(DT)$, $|\xi_a| = |\xi_b| = |\xi|$;
3. *if* $\exists f \in DF$, *such that* $a \in f$ *and* $|f| > 1$. *If* $\exists b \in (f - \{a\})$ *and* $P_{DF}(b) > 1$, *then* $\forall \xi_a \in I(a)$ *and* $\forall \xi_b \in I(b)$, $|\xi_a| \ge |\xi_b|$. *where* $I(a)$ *denotes the set of the shortest implicants that include attribute a*, $P_{DF}(a)$ *denotes the occurrence frequency of a in discernibility function* DF.

Proof Suppose that the notation $SPI(F)$ denotes the set of the shortest prime implicants of Boolean function F.

1. $|f| = 1$ means that the clause f only includes single attribute a, so a is included in core set. According to theorem 3, a will be included in all prime implicants of discernibility function DF, so $I(a) = SPI(DF) = MR(DT)$.
2. Since $\forall e \in f$, $P_{DF}(e) = 1$, so $\forall f' \in (DF - f)$, $f \cap f' = \emptyset$. Then $\forall \xi_e \in I(e)$, it has $|\xi_e| = 1 + |\xi'|$, where $\xi' \in SPI(DF - f)$. Further, $\forall \xi \in MR(DT)$, $\xi \cap f = 1$, so $\xi = \xi' \cup \{e\}$, thus $\forall \xi_a \in I(a)$ and $\forall \xi_b \in I(b)$, it has $|\xi_a| = |\xi_b| = |\xi|$.
3. Suppose $P_{DF}(b) = 2$ (the other situations $P_{DF}(b) > 2$ can be illustrated similarly), so $\exists f' \in (DF - f)$ such that $b \in f'$ and $a \notin f'$. $\forall \xi_a \in I(a)$, $\forall \xi_b \in I(b)$, $\forall \xi' \in SPI(DF - f)$ and $\forall \xi'' \in SPI(DF - f - f')$, it has $|\xi_a| = 1 + |\xi'|$, $|\xi_b| = 1 + |\xi''|$, according to theorem 4, we have $|\xi''| \le |\xi'|$, thus $|\xi_a| \ge |\xi_b|$.

□

Theorem 5 shows that the search paths beginning from the attributes which occurrence frequencies in the discernibility function are equal to one need not to be considered in the global search process. Because these attributes either must be included in a minimal reduct (as (1), (2) in Theorem 5), or may be not included in any minimal reducts (as (3) in Theorem 5). So superfluous search works can be avoided and the efficiency of global search can be improved.

In order to find a minimal reduct of a decision table, an iterative algorithm can be constructed by utilizing Theorems 2 and 3 repeatedly. Based on Theorems 2, 3, 4 and 5, some search strategies can be added to the process of minimal attribute reduction based on depth-first search method.

Depth search strategy one. Which attribute will be chosen as a decomposition variable is very important in the search process. The attributes will be chosen according to their significance from high to low. Because choosing an attribute with higher significance will reduce search space fast. In a depth search path, if attribute a has been chosen as a decomposition variable for Boolean function DF_k in the kth step, then we can only deal with $DF_{k+1} = DF_k - \{f | a \in f, f \in DF_k\}$ in the $(k+1)$th step according to Theorem 3.

Depth search strategy two. If the extended order of variables established at the first time will not be changed in the sequel decomposition procedures, then the order is called as static variable order. On the contrary, if attribute significance will be changed dynamically based on different Boolean function in the sequel decomposition and the related extended order is also adjusted simultaneously, then the order is called as dynamic variable order. During the implementation of the algorithm, the later will be applied.

Depth search strategy three. If the length of current attribute sequence in a depth search path is equal to the length of candidate minimal reduct, then the current depth search is terminated, and the path turns back to the upper layer for width searching right along.

Width search strategy. Suppose $\Omega_{DF} = \{a_1, a_2, \ldots, a_t\}$ and the variable order is $a_1 > a_2 > \cdots > a_t$ for the first decomposition. According to depth search strategy one, a_k is preferential to a_{k+1}. After search path beginning from a_k terminated, a shortest implicant that includes a_k has been found. For search path beginning from a_{k+1}, it just need to deal with Boolean function $\wedge\{f|f \in DF \wedge a_k \notin f\} \wedge \{(f - \{a_k\})|f \in DF \wedge a_k \in f\}$ using Theorems 2 and 3 iteratively. If there is a clause which becomes to empty after removing some variables during decomposition procedures, then the algorithm turns back to the upper layer.

The complete algorithm for minimal attribute reduction based on discernibility function (*CAMARDF*) can be described as follows:

Algorithm 2: *CAMARDF*

```
Input:  Decision table S = (U, C ∪ D, V, ρ);
Output:  a minimal reduct of S preserving the property Δ.
Initialization: Reduct.length=0, MinReduct.length=|C| and simplified
discernibility function DF has been constructed well.
    CAMARDF( DF )
    {
1       ComputeSIG( a , a ∈ Ω_DF );
2       SortSIG( sig(a) , a ∈ Ω_DF );
3       i=0;
4       do{
5           Reduct.length++;
6           if( Reduct.length = MinReduct.length ){
7               Reduct.length−−;
8               return;
9           }//end if

10          if( i>0 ){
11              DF = DF\{Attribute[i − 1]};
12              if( ∃f_j ∈ DF, f_j = ∅ ){
13                  Reduct.length−−;
14                  return;
15              }//end if
16          }//end if

17          Reduct=Reduct∪Attribute[i];
18          DF' = DF − {f_j|f_j ∈ DF ∧ Attribute[i] ∈ f_j};
19          if( DF' = ∅ ){
20              if( MinReduct.length > Reduct.length )
21                  MinReduct=Reduct
22          }//end if
23          else
24              CAMARDF( DF' );

25          Reduct=Reduct - Attribute[i];
26          Reduct.length−−;
27          i++;
28      }while( sig(Attribute[i]) > 1 ∧ i < |C| );
29  }//end CAMARDF
```

where $DF \backslash \{Attribute[i-1]\}$ denotes $\forall f_j \in DF$, if $Attribute[i-1] \in f_j$, then $f_j = f_j - \{Attribute[i-1]\}$.

Reduct and *MinReduct* are global variables in Algorithm 2. The attribute sequence in current depth search path is saved in *Reduct*, the current candidate minimal reduct is saved in *MinReduct* and the last *MinReduct* is the optimal result that will be obtained. The operation ComputeSIG in line 1 computes the significance for each variable in discernibility function *DF*. The significance of attributes in *CAMARDF* is measured by their occurrence frequencies in Boolean functions during decomposition procedures. The operation SortSIG in line 2 sorts variables from high to low based on their significance, and the variable order is saved in array *Attribute*. These two steps are corresponding to the depth search strategy two.

The completeness of the algorithm *CAMARDF* for minimal attribute reduction can be guaranteed by Theorem 2. In the implementation, line 5 to line 9, line 10 to line 16, line 18 to line 24 are corresponding to the depth search strategy three, the width search strategy, the depth search strategy one, respectively, and the terminal constrains of do-while instruction reflect Theorem 5.

The entire process for minimal attribute reduction based on discernibility function can be schematically represented in Fig. 3.

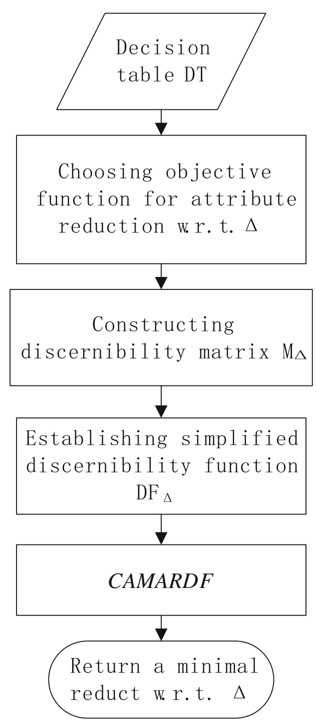

Fig. 3 The entire process for minimal attribute reduction

For Table 3, its relevant discernibility function under positive region reduct definition can be obtained as follows:

$$DF = (a_1 \vee a_3) \wedge (a_1 \vee a_2 \vee a_4) \wedge (a_1 \vee a_4 \vee a_8) \wedge (a_1 \vee a_2 \vee a_9) \wedge (a_3 \vee a_7 \vee a_{12}) \wedge (a_3 \vee a_6 \vee a_{13}) \wedge (a_2 \vee a_5) \wedge (a_4 \vee a_{10}) \wedge a_{11}.$$

After Johnson's strategy or its improved version Semi-Minimal reduct algorithm (Nguyen and Nguyen 1996) is applied, the attribute reduct is formed as $\{a_1, a_3, a_2, a_4, a_{11}\}$ which is not a proper minimal reduct. It also demonstrates that the approximate algorithms are incomplete for minimal attribute reduct problem.

According to the occurrence frequency of each variable in *DF*, the extended order of variables for the first decomposition in the algorithm *CAMARDF* is given as:

$$a_1 > a_2 > a_3 > a_4 > a_5 > a_6 > a_7 > a_8 > a_9 > a_{10} > a_{11} > a_{12} > a_{13}.$$

The candidate minimal reduct can be gradually computed as follows:

$\{a_1, a_3, a_2, a_4, a_{11}\} \in I(a_1)$, such that $Reduct.length = 5 < MinReduct.length = |C|$, $MinReduct = \{a_1, a_3, a_2, a_4, a_{11}\}$;

$\{a_2, a_3, a_4, a_{11}\} \in I(a_2)$, such that $|\{a_2, a_3, a_4, a_{11}\}| < MinReduct.length = 5$, $MinReduct = \{a_2, a_3, a_4, a_{11}\}$.

Since $Reduct.length = MinReduct.length$ when the search paths begin from a_3 and a_4, the candidate minimal reduct *MinReduct* will not be changed according to depth search strategy three.

For any other variables, their occurrence frequencies in Boolean function *DF* are all equal to one, the search paths beginning from them cannot be considered. So the final obtained minimal reduct is $\{a_2, a_3, a_4, a_{11}\}$ which is not found by Algorithm 1. During the implementation, the core attribute set can be firstly computed in each decomposition process, however the total consumed time is almost as the same as in Algorithm 2.

6 Experimental results

The algorithm *CAMARDF* was run on a personal computer with Intel Pentium Dual-Core E2140 1.6 GHz processor and 1 Gb memory. The operating system is Windows XP and the programs are implemented using VC 6.0.

Twelve consistent UCI data sets are chosen to test the proposed algorithm. The objective function for attribute reduction that preserves the positive region is only considered. In order to illustrate the reduct that is found by the proposed algorithm is the minimal one, the set of all reducts of each data set are computed by using attribute reduction algorithm based on algebraic equations (Miao et al. 2010). The discernibility function and the reducts of each data set are shown in Table 4.

In Table 4, Num, Max, Min and Avg denote the number of clauses, the maximal length of clauses, the minimal length of clauses and the average length of clauses, respectively. Core denotes the number of core attributes for data sets. MR means the minimal reducts of data sets. MaxL and MinL denote the maximal length and the minimal length of reducts, respectively.

Only the first 10,000 objects of the whole connect data set and only the first 200 objects of the original DNA data set (StatLog version) are chosen. We just want to test the

Table 4 The discernibility functions and reducts for UCI data sets

Data sets	No. of objects	No. of attributes	Clauses				Reducts				
			Num	Max	Min	Avg	Core	No. of reducts	No. of MR	MaxL	MinL
zoo	101	17	14	6	1	3	2	33	7	7	5
breast	699	10	19	5	1	3	1	20	8	5	4
mushroom	8,124	23	30	12	2	6	0	292	13	8	4
chess	3,196	37	29	2	1	1	27	4	4	29	29
tic-tac-toe	958	10	36	2	2	2	0	9	9	8	8
soy	47	36	99	14	6	9	0	756	4	8	2
audiology	200	70	202	10	1	5	3	113,329	4	31	12
connect	10,000	43	440	2	2	2	0	32	9	36	25
led24-1	200	25	2,458	12	3	8	0	66,800	95	15	11
led24-2	2,000	25	371	6	1	3	3	495	29	20	18
led24-3	10,000	25	23	1	1	1	23	1	1	23	23
DNA	200	61	11,760	53	30	44	0	–	–	–	5

Table 5 The consumed time of searching a minimal reduct for UCI data sets

Data sets	No. of objects	No. of attributes	No. of clauses	Time		
				DF	*CAMARDF*	Total
zoo	101	17	14	0.015	0	0.015
breast	699	10	19	0.256	0	0.256
mushroom	8,124	23	30	130.062	0	130.062
chess	3,196	37	29	7.593	0	7.593
tic-tac-toe	958	10	36	0.531	0	0.531
soy	47	36	99	0.015	0	0.015
audiology	200	70	202	0.468	2.094	2.562
connect	10,000	43	440	343.984	0.047	344.031
led24-1	200	25	2,458	6.281	4.922	11.203
led24-2	2,000	25	371	82.468	0.157	82.625
led24-3	10,000	25	23	187.421	0	187.421
DNA	200	61	11,760	70.890	25.594	96.484

The measurement of time is s and zero denotes that time is less than 0.001 s.

algorithm for minimal attribute reduction, so it is not necessary to choose all objects since it has to spend more time to get all reducts of these two original data sets. Even so, there were still too many reducts for the chosen DNA data set and the memory was not sufficient.

The result, which is obtained by *CAMARDF* for each data set, is a real minimal reduct according to the set of all reducts of each data set. For DNA data set, it cannot get all reducts, but a minimal reduct can be obtained by the proposed algorithm. From the data presented in Table 4, the number of minimal reducts is significantly smaller compared with the total number of reducts. Furthermore, the length of a minimal reduct is much shorter compared with the number of entire condition attributes. It indicates only several attributes can be used to describe original data sets without losing the property under consideration. Then, the data set can be compressed as much as possible under its minimal reducts, and the rule set will be more concise, general and understandable. The time for searching a minimal reduct of a given data set by the proposed algorithm is presented in Table 5.

The number of clauses of connect data set is more than audiology data set, but the average length of clauses in connect data set is shorter than in audiology data set. The result is that the consumed time of *CAMARDF* for connect data set is smaller than audiology data set. It implies that the consumed time of *CAMARDF* is not only related to the number of clauses in *DF* but also closely related to the average length of clauses.

The clause numbers of discernibility functions in UCI data sets are small (only except DNA data set in Table 5). In order to test the efficiency of *CAMARDF* further, some synthetic data sets are exploited which discernibility functions include more clauses. Attribute values are generated randomly between 0 and 9. The number of attributes and objects varies from 50 to 95 in step of 5, respectively. Ten synthetic data sets can be produced. The last attribute column in each data set is chosen as a decision attribute. The results of synthetic data sets are provided in Table 6.

There are no perceptible relations between objects in a given synthetic data set due to the attribute values are generated randomly. The clause numbers of discernibility functions will increase when increasing the number of objects or the number of attributes. In Table 6, it can be found that the average length of clauses in each data set approaches to the number of entire attributes of this data set, namely, each clause of the discernibility function includes the most attribute variables. During the implementation of *CAMARDF*, when a decomposition variable is considered, the search space is reduced fast and current depth search path will be terminated quickly since the clauses that include this variable should be eliminated. During the experiments, it cannot obtain the set of all reducts when the size of the given synthetic data set is more than DT_3 in Table 6. However, a minimal reduct can be rapidly attained with the proposed method.

As shown in Tables 5 and 6, the main time for searching a minimal reduct is spent on establishing simplified discernibility function *DF*. The minimal reduct problem can be quickly solved by the proposed notion after associated *DF* being computed well. However, the time for establishing the simplified discernibility function is related to the number of objects and the number of attributes, namely, the structural complexity of data sets.

Table 6 The discernibility functions and consumed time for synthetic data sets

Data sets	No. of objects	No. of attributes	Clause				Time		
			Num	Max	Min	Avg	*DF*	*CAMARDF*	Total
DT_1	50	50	831	46	35	42	0.578	0.031	0.609
DT_2	55	55	1144	51	39	47	0.968	0.047	1.015
DT_3	60	60	1371	56	42	51	1.671	0.063	1.734
DT_4	65	65	1687	60	48	56	2.187	0.078	2.265
DT_5	70	70	2018	66	48	61	3.812	0.125	3.937
DT_6	75	75	2350	71	55	65	4.515	0.172	4.687
DT_7	80	80	2744	76	60	70	7.343	0.219	7.562
DT_8	85	85	3087	81	59	74	8.109	0.266	8.375
DT_9	90	90	3465	85	66	78	11.984	0.344	12.328
DT_{10}	95	95	3912	90	72	83	13.89	0.438	14.328

7 Conclusion

A reduct is a special attribute subset that can preserve a certain property under consideration. According to the diversified properties of a decision table, some objective functions for attribute reduction have been put forward. Essentially, only six and two different types of alternative objective functions for attribute reduction are in inconsistent and consistent decision tables, respectively. The revealed relationship among 13 typical kinds of objective functions for attribute reduction will be beneficial for designing new reduct definitions and developing more efficient algorithms.

Though available heuristic algorithms for attribute reduction have high efficiency, it is found that their results are not optimal and even not the real reducts. According to the characteristics of discernibility functions, a complete algorithm for searching a proper minimal reduct is proposed. Since it is built based on reasoning mechanisms, the proposed algorithm is general for all complete decision tables just after the corresponding discernibility function is constructed well, no matter which objective functions for attribute reduction will be applied. The rule sets derived under different attribute reduction objective functions are diversified. How to quantitatively measure and compare them could be a subject of further studies.

Acknowledgments The authors are grateful to the anonymous referees for their valuable comments and suggestions. This work is supported by National Natural Science Foundation of China (Serial Nos. 60475019, 61075056, 60970061) and the research fund for the Doctoral Program of Higher Education in China (Serial No. 20060247039).

References

An A, Shan N, Chan C et al (1996) Discovering rules for water demand prediction: an enhanced rough set approach. Eng Appl Artif Intell 9(6):645–653

Asuncion A, Newman DJ (2007) UCI machine learning repository. University of California, Irvine, CA. http://archive.ics.uci.edu/ml/

Beynon M, Peel MJ (2001) Variable precision rough set theory and data discretization: an application to corporate failure prediction. Int J Manag Sci Omega 29:561–576

Chang LY, Wang GY, Wu Y (1999) An approach for attribute reduction and rule generation based on rough set theory. J Softw 10(11):1206–1211 (in Chinese)

Deng DY, Huang HK, Li XJ (2007) Comparison of various types of reductions in inconsistent systems. Acta Electron Sin 35(2):252–255 (in Chinese)

Dimitras AI, Slowinski R, Susmaga R et al (1999) Business failure using rough sets. Eur J Oper Res 114:263–280

Hu XH, Cercone N (1995) Learning in relational databases: a rough set approach. Int J Comput Intell 11(2):323–338

Jelonek J, Krawiec K, Slowinski R (1995) Rough set reduction of attributes and their domains for neural networks. Int J Comput Intell 11(2):339–347

Komorowski J, Pawlak Z, Polkowski L, Skowron A (1999) Rough sets a tutorial. In: Pal SK, Skowron A (eds) Rough fuzzy hybridization: a new trend in decision-making. Springer, Singapore, pp 3–98

Kryszkiewicz M (1998) Rough set approach to incomplete information systems. Inform Sci 112:39–49

Kryszkiewicz M (2001) study of alternative types of knowledge reduction in inconsistent systems. Int J Intell Syst 16:105–120

Li DY, Zhang B (2004) On knowledge reduction in inconsistent decision information systems. Int J Uncertain Fuzziness Knowl Based Syst 12(5):651–672

Liang JY, Shi ZZ, Li DY et al (2006) Information entropy, rough entropy and knowledge granulation in incomplete information systems. Int J Gen Syst 35(6):641–654

Liang JY, Wang BL, Qian YH et al (2008) An algorithm of constructing maximal consistent blocks in incomplete information systems. Int J Comput Sci Knowl Eng 2(1):11–18

Leung Y, Li DY (2003) Maximal consistent block technique for rule acquisition in incomplete information systems. Inform Sci 153:85–106

Mi JS, Wu WZ, Zhang WX (2004) Approaches to knowledge reduction based on variable precision rough set model. Inform Sci 159:255–272

Miao DQ, Wang J (1997) Information-based algorithm for reduction of knowledge. In: Proceedings of the 1997 IEEE international conference on intelligent processing systems (ICIPS'97), 28–31 October, Beijing, China, pp 1155–1158

Miao DQ, Zhao Y, Yao YY et al (2009) Relative reducts in consistent and inconsistent decision tables of the Pawlak rough set model. Inform Sci 179:4140–4150

Miao DQ, Zhou J, Zhang N et al (2010) Research of attribute reduction based on algebraic equations. Acta Electron Sin 38(5):1021–1027 (in Chinese)

Nguyen SH, Nguyen HS (1996) Some efficient algorithms for rough set methods. In: Proceedings of the sixth international conference on information processing and management of uncertainty in knowledge-based systems (IPMU'96), 1–5 July, Granada, Spain, pp 1451–1456

Nguyen HS, Slezak D (1999) Approximation reducts and association rules correspondence and complexity results. In: Zhong N, Skowron A, Oshuga S (eds) Proceedings of RSFDGrC'99, Yamaguchi, Japan, LNAI 1711, pp 137–145

Pawlak Z (1982) Rough sets. Int J Comput Inform Sci 11(5):341–356

Pawlak Z, Wong SKM, Ziarko W (1988) Rough sets: probabilistic versus deterministic approach. Int J Man Mach Stud 29:81–95

Pawlak Z, Skowron A (2007) Rudiments of rough sets. Inform Sci 177(1):3–27

Pedrycz W (2007) Granular computing—the emerging paradigm. J Uncertain Syst 1(1):38–61

Pedrycz W, Skowron A, Kreinovich V (2008) Handbook of granular computing. John Wiley & Sons Inc., New York

Qian YH, Liang JY, Wang F (2009) A new method for measuring the uncertainty in incomplete information systems. Int J Uncertain Fuzziness Knowl Based Syst 17(6):855–880

Qin KY, Pei Z, Du WF (2005) The relationship among several knowledge reduction approaches. FSKD 2005, LNAI 3613, pp 1232–1241

Skowron A, Rauszer C (1991) The discernibility matrices and functions in information systems. In: Slowinski R (eds) Intelligent decision support handbook of applications and advances of the rough sets theory. Kluwer Academic Publishers, Dordrecht, pp 331–362

Slezak D (2000) Various approaches to reasoning with frequency based decision reducts: a survey. In: Rough set methods and applications: new developments in knowledge discovery in information systems. Physica-Verlag Gmbh, Heidelberg, pp 235–285

Starzyk JA, Nelson DE, Sturtz K (2000) A mathematical foundation for improved reduct generation in information systems. Knowl Inform Syst 2:131–146

Thangavel K, Pethalakshmi A (2009) Dimensionality reduction based on rough set theory: a review. Appl Soft Comput 9:1–12

Wang CZ, Wu CX, Chen DG (2008) A systematic study on attribute reduction with rough sets based on general binary relations. Inform Sci 178:2237–2261

Wang GY, Yu H, Yang DC (2002) Decision table reduction based on conditional information entropy. Chin J Comput 25(7):759–766 (in Chinese)

Wang GY (2003) Rough reduction in algebra view and information view. Int J Intell Syst 18:679–688

Wang GY, Zhao J, Wu J (2005) A comparative study of algebra viewpoint and information viewpoint in attribute reduction. Fund Inform 68(3):289–301

Wang J, Miao DQ (1998) Analysis on attribute reduction strategies of rough set. J Comput Sci Technol 13(2):189–193

Wang J, Wang J (2001) Reduction algorithms based on discernibility matrix: the order attributes method. J Comput Sci Technol 16(6):489–504

Wong SKM, Ziarko W (1985) On optimal decision rules in decision tables. Bull Polish Acad Sci 33(11–12):693–696

Xu ZY, Liu ZP, Yang BR et al (2006) A quick attribute reduction algorithm with complexity of $\max(O(|C||U|), O(|C|^2|U/C|))$. Chin J Comput 29(3):391–399 (in Chinese)

Xu ZY, Yang BR, Song W et al (2008) Comparative research of different attribute reduction definitions. J Chin Comput Syst 5(5):848–853 (in Chinese)

Yao YY, Zhao Y, Wang J (2006) On reduct construction algorithms. In: Proceedings of the first international conference on rough sets and knowledge technology (RSKT'06), 24–26 July, Chongqing, China, pp 297–304

Yao YY, Zhao Y (2009) Discernibility matrix simplification for constructing attribute reducts. Inform Sci 179(7):867–882

Yao YY, Zhao Y, Wang J et al (2008) A model of user-oriented reduct construction for machine learning. Transactions on rough sets VIII. Springer, Berlin, LNCS 5084, pp 332–351

Yao YY, Zhao Y (2008) Attribute reduction in decision-theoretic rough set models. Inform Sci 178:3356–3373

Ye DY, Chen ZJ (2006) An improved discernibility matrix for computing all reducts of an inconsistent decision table. In: Proceedings of the fifth IEEE international conference on cognitive informatics (ICCI'06), 17–19 July, Beijing, China, pp 305–308

Zakowski W (1993) Incomplete information systems. Bull Polish Acad Sci Tech Sci 41(2):175–181

Zhang WX, Mi JS, Wu WZ (2003) Approaches to knowledge reductions in inconsistent systems. Int J Intell Syst 18:989–1000

Zhao Y, Yao YY (2007) Data analysis based on discernibility and indiscernibility. Inform Sci 177:4959–4976

Zhao Y, Luo F, Wong SKM, Yao YY (2007) A general definition of an attribute reduct. In: Proceedings of the second rough sets and knowledge technology (RSKT'07), 14–16 May, Toronto, Canada, pp 101–108