Information Sciences 536 (2020) 298–316

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Improved general attribute reduction algorithms

Baizhen Li [a], Zhihua Wei [a,*], Duoqian Miao [a], Nan Zhang [b], Wen Shen [a], Chang Gong [a], Hongyun Zhang [a], Lijun Sun [a]

[a] Tongji University, Shanghai 201804, PR China
[b] Yantai University, Yantai, Shandong 264005, PR China

ARTICLE INFO

Article history:
Received 7 December 2019
Received in revised form 7 May 2020
Accepted 11 May 2020
Available online 16 May 2020

MSC:
00–01
99–00

Keywords:
Attribute reduction
Granular computing
Rough sets

ABSTRACT

Attribute reduction is a critical issue in rough sets theory. In recent years, there are many kinds of attribute reduction proposed, such as positive region preservation reduction, generalized decision preservation reduction, distribution preservation reduction, maximum distribution preservation reduction, and relative discernibility relation preservation reduction. General reduction approaches to obtaining various types of reducts also have been explored, but they are computationally time-consuming in the condition of large-scale data processing. In this study, we focus on the efficient general reduction algorithm to obtain five typical reducts mentioned above. At first, we introduce a concept called granularity space to establish a unified representation of five typical reducts. Based on the unified representation, we construct two quick general reduction algorithms by extending the positive region approximation to the granularity space. Then, we conduct a series of comparisons with existing reduction algorithms in aspects of theoretical analysis and experiments to evaluate the performance of the proposed algorithms. The results of analysis and experiments indicate that the proposed algorithms are effective and efficient.

1. Introduction

Rough sets theory, introduced by Z. Pawlak [24] in 1982, is an efficient tool for imprecise, incomplete and uncertain information processing [11,28,47]. Currently, rough sets theory has been successfully applied to many practical problems, including machine learning [1,40], pattern recognition [9,10], data mining [35], decision support systems [15], etc.

Attribute reduction is one of the core concepts in rough sets[32]. It represents the process of obtaining attribute reduct, *i.e.*, a minimal set of attributes that can preserve the same ability of classification as the entire attribute set. Main studies of attribute reduction can be classified into two categories: the appropriate definition of attribute reduction and the efficient reduction algorithm.

The appropriate definition of attribute reduction is a prerequisite for the good performance of attribute reducts in classification. After analyzing the relation of the positive region and the classification rule in consistent decision tables, Pawlak proposed the positive region preservation reduction [24]. Kryszkiewicz proposed two types of reduction for inconsistent decision tables: the generalized decision preservation reduction and the distribution preservation reduction [17], which guarantee the property of possible decisions of objects and the decision class membership distribution of objects unchanged

* Corresponding author.
E-mail address: zhihua_wei@tongji.edu.cn (Z. Wei).

https://doi.org/10.1016/j.ins.2020.05.043
0020-0255/

respectively. After that, Zhang et al. [48] presented the maximum distribution preservation reduction as a compromise between the capability of the generalized decision preservation reduction and the complexity of the distribution preservation reduction. Thereafter, it emerged as a mainstream that researchers design appropriate attribute reduction definitions based on understanding the relationship between different attribute reductions. Liu et al. [21] presented the distribution preservation reduction, the maximum distribution preservation reduction, and the generalized decision preservation reduction in the way of the classic reduction. Furthermore, Ref [23] classified the existing reduction into three types: the region preservation reduction, the decision preservation reduction, and the relationship preservation reduction. Meanwhile, the relative discernibility relation preservation reduction was proposed. Then, Zhou et al. [49] reviewed the existing attribute reduction, and concluded that there were six different types of attribute reduction for complete inconsistent decision tables. On this basis, Jia et al. [14] explored the reduction definition from the user's perspective to alleviate the difficulties of choosing appropriate attribute reduction for specific applications. After reviewing the discernibility relation of different reducts, Ge et al. [6] proposed a unified definition of five types of attribute reduction.

The efficient reduction algorithm [18,22,38] is the central focus of researchers' studies. Attribute reduction algorithms can be grouped into two classes [33]: the discernibility matrix-based algorithm [29,30] and the heuristic algorithm. Many researchers studied the discernibility matrix-based attribute reduction algorithms because it is easily understandable and can find all reducts [3,17,23,37,41,44,48,49]. However, the discernibility matrix-based method is computationally expensive. Therefore, heuristic approaches are applied to attribute reduction processes. The heuristic approach is composed of two parts: the heuristic function and the search strategy [45]. The heuristic function is the fitness function of a heuristic approach. Existing definitions of heuristics are mainly based on three aspects: dependency degree [8], entropy [31,34,43], and consistency [2,7,19]. The search strategy is the control structure of the heuristic approach. There are two basic search strategies in heuristic approaches [36]: the directional search strategy and the non-directional search strategy. The directional search strategy contains three kinds of methods: the deletion method, the addition method, and the addition-deletion method, and it has been applied in mainstream heuristic reduction algorithms. The non-directional search strategy is usually applied in evolutionary algorithms [4,13,16] and some optimization methods [12,46]. To further increase the computational efficiency, many researchers studied acceleration mechanisms of the heuristic attribute reduction method. Ref [6,42] computed equivalence classes using a classic sort algorithm, which improved the speed of attribute reduction algorithms. Qian et al. [26] presented a counting sort algorithm to reduce the computation cost of positive regions and core attributes. Furthermore, Qian et al. [27] studied an acceleration strategy for the positive region preservation reduction and three types of entropy reductions. Liang et al. [20] developed a new accelerator that simultaneously decreased the size of the universe and the number of attributes in each iteration process of attribute reduction.

There are five types of representative reduction for complete inconsistent decision tables, *i.e.*, positive region preservation reduction, generalized decision preservation reduction, distribution preservation reduction, maximum distribution preservation reduction, and relative discernibility relation preservation reduction. Ref [14] explored the general attribute reduction definition and Ref [6] researched the approaches to obtaining those five typical reducts. However, existing general reduction algorithms are computationally time-consuming in processing large-scale data due to the lack of an efficient framework of reduction theory. To alleviate this problem, we propose a new unified representation of five typical reducts, and on this basis, we propose two quick general reduction algorithms.

Firstly, to construct a new definition of general attribute reduction, we introduce the concept of granularity space and analyze the properties of its binary relation. By associating the indiscernibility relation of granularity space with the indiscernibility relation of five reducts, we construct the definition of general attribute reducts in the way of granularity space. Finally, by extending the positive region approximation to granularity space, we develop two quick general reduction algorithms. Meanwhile, a series of analyses in aspects of theory and experiments are conducted to evaluate the effectiveness and efficiency of proposed algorithms. Two major contributions of this study are listed as follows. (1) We introduce a concept named granularity space to represent five attribute reductions in a unified framework; (2) We extend the positive region approximation to the granularity space and design a new acceleration strategy for general attribute reduction algorithms, which can expand the acceleration domain from the positive region to the universe of decision tables.

The rest of this paper is organized as follows. In Section 2, we briefly review preliminary notions related to five types of representative reduction and the classic definition of the general reduct. Besides, we discuss the process of the discernibility matrix-based reduction method. In Section 3, we analyze the room for improvement in the classic general reduct definition and propose the granularity space as an example to support the analysis. After that, we present the quick general heuristic reduction algorithms based on granularity space. Besides, we explain the advantage of proposed algorithms and their relationship to existing reduction algorithms. In Section 4, we conduct a series of experiments with several UCI data sets to evaluate the performance of proposed reduction algorithms. Finally, Section 5 concludes this paper and brings some remarks about the work of this paper.

2. Preliminaries

In this part, we briefly review the concepts of decision tables, the notions of five representative attribute reducts, and the definition of general reducts. Next, we brush up on the discernibility matrix-based reduction algorithm, which is the kin of two proposed algorithms.

The research object of rough sets theory is called the information system, which can be expressed as a four-tuple, *i.e.*, (U, A, V, f). Here U stands for the universe of discourse, a non-empty finite set of instances. A is the set of attributes, $V = \bigcup_{a\in A} V_a$ is the set of all attribute values, and $f : U \times A \to V$ is an information function that maps an object in U to exactly one value in V_a. For $x \in U, a \in A$, we have $f(x, a) \in V_a$. In the classification problem, the information system contains two kinds of attributes, and it can be characterized by a decision table $DT = (U, C \cup D, V, f)$ with $C \cap D = \varnothing$, where an element of C is called a condition attribute, C is called the condition attribute set, an element D is called a decision attribute, and D is called the decision attribute set [20].

For the condition attribute set $B \subseteq C$, the indiscernibility relation $\mathrm{IND}(B)$ is defined by $\mathrm{IND}(B) = \{\langle x, y\rangle \,|\, x, y \in U, f(x, a) = f(y, a), \forall a \in B\}$. For an instance $x \in U$, the equivalence class of x, being represented as $[x]_B$, is described by $\{y \,|\, y \in U, \langle x, y\rangle \in \mathrm{IND}(B)\}$. The family of all equivalence classes of $\mathrm{IND}(B)$, *i.e.*, the partition determined by B, is denoted by $U/\mathrm{IND}(B)$ or simply U/B. X, a non-empty subset of U, is called a concept of U. The B-lower approximation $\underline{B}(X)$ and the B-upper approximation $\overline{B}(X)$ of the concept X are respectively defined by $\underline{B}(X) = \{x \in U \,|\, [x]_B \subseteq X\}$ and $\overline{B}(X) = \{x \in U \,|\, [x]_B \cap X \neq \varnothing\}$. The classification ability of conditional attribute C is measured by the relation between $\mathrm{IND}(C)$ and $\mathrm{IND}(D)$, and there are some uncertain situations that objects x, y with the same value of conditional attributes perform differently in the decision attributes. The uncertain situations are represented as the difference between two notions, *i.e.*, positive region and boundary region which are induced from indiscernibility relation. The B-positive region $\mathrm{POS}_B(D)$ and B-boundary region $\mathrm{BND}_B(D)$ are defined as

$$\mathrm{POS}_B(D) = \bigcup_{X\in U/D} \underline{B}(X),$$
$$\mathrm{BND}_B(D) = \bigcup_{X\in U/D} \left(\overline{B}(X) - \underline{B}(X)\right).$$

$\mathrm{POS}_B(D)$ consists of objects which perform consistently in decision; $\mathrm{BND}_B(D)$ is comprised of objects which perform inconsistently in decision. Generally, we take the difference between them as a measurement of the classification ability of B. In this viewpoint, an attribute set $B \subseteq C$ satisfying $\mathrm{POS}_B(D) = \mathrm{POS}_C(D)$ is meaningful for feature selection and knowledge representation. Exactly speaking, B, an attribute set satisfying $\mathrm{POS}_B(D) = \mathrm{POS}_C(D) \wedge (\forall B' \subset B, \mathrm{POS}_{B'}(D) \neq \mathrm{POS}_C(D))$, is called positive region preservation attribute reduct. After proposing the positive region preservation attribute reduct, many extensions of that are investigated. Considering the focus of paper, we list five typical reducts' definitions summarized in Ref [49] here.

Definition. Given a decision table $DT = (U, C \cup D, V, f)$,

(1) B is a positive region preservation reduct (denoted as PRPR) of C with respect to D if B satisfies $\mathrm{POS}_B(D) = \mathrm{POS}_C(D)$ and $\forall B' \subset B,\ \mathrm{POS}_{B'}(D) \neq \mathrm{POS}_C(D)$;
(2) B is a generalized decision preservation reduct (denoted as GDPR) of C with respect to D if B satisfies $\forall x \in U, \delta_B(x) = \delta_C(x)$ and $\forall B' \subset B, \exists x \in U, \delta_{B'}(x) \neq \delta_B(x)$, where $\delta_B(x) = \{f(y, D) \,|\, x \in U \wedge y \in [x]_B\}$;
(3) B is a distribution preservation reduct (denoted as DPR) of C with respect to D if B satisfies $\forall x \in U, \mu_B(x) = \mu_C(x)$ and $\forall B' \subset B, \exists x \in U, \mu_{B'}(x) \neq \mu_B(x)$, where $\mu_B(x) = \left(P(D_1|[x]_B), P(D_2|[x]_B), \cdots, P(D_{|U/D|}|[x]_B)\right), P(D_j|[x]_B) = \frac{|D_j\cap[x]_B|}{|[x]_B|}, x \in U, D_j \in U/D (j = 1, 2, \cdots, |U/D|)$;
(4) B is a maximum distribution preservation reduct (denoted as MDPR) of C with respect to D if B satisfies $\forall x \in U, \phi_B(x) = \phi_C(x)$ and $\forall B' \subset B, \exists x \in U, \phi_{B'}(x) \neq \phi_B(x)$, where $\phi_B(x) = \left\{D_j | \frac{|[x]_B\cap D_j|}{|[x]_B|} = \max_{k=1}^{|U/D|}\left\{\frac{|[x]_B\cap D_k|}{|[x]_B|}\right\}\right\}$;
(5) B is a relative discernibility relation preservation reduct (denoted as DRPR) of C with respect to D if B satisfies $\mathrm{IND}(B|D) = \mathrm{IND}(C|D)$ and $\forall B' \subset B, \mathrm{IND}(B'|D) \neq \mathrm{IND}(B|D)$, where $\mathrm{IND}(B|D) = \{\langle x, y\rangle \in U \times U \,|\, \wedge (\forall a \in B \to f(x, a) = f(y, a)) \vee f(x, D) = f(y, D)\}$.

Those five reducts can be taken as the special cases of the general reduct proposed by Yao et al. [45], which can be written as follows.

Definition. Given a decision table $DT = (U, C \cup D, V, f)$ and a certain property $\mathbb{P}$ of DT, an attribute set $B \subseteq C$ is called a reduct of C if it satisfies the following three conditions:

(1) Evaluability condition: the property can be represented by an evaluation function $e : 2^C \to (L, \preceq)$;
(2) Jointly sufficient condition: $e(A) \preceq e(B)$;
(3) Individually necessary condition: for any $B' \subset B$, $\neq g(e(A \preceq e(B')))$.

Here $e : 2^C \to (L, \preceq)$ is an evaluation or fitness function, which maps an attribute set to an element of a poset L equipped with the partial order relation $\preceq$, *i.e.*, $\preceq$ is relexive, anti-symmetric and transitive. Attribute reduction algorithms, approaches to obtaining a reduct or reducts, can be classified into two groups: the discernibility matrix-based algorithm and the heuris-

tic algorithm. For the limitation of paper focus, here we only review the discernibility matrix-based algorithm. A general discernibility matrix given by Miao et al. [23] is defined as follows.

Definition. Given a decision table $DT = (U, C \cup D, V, f)$, its discernibility matrix $DM = (DM(x,y))$ is a $|U| \times |U|$ matrix. $DM(x,y)$ for an object pair (x,y) is $\{a \,|\, \langle x,y\rangle \in \mathrm{DIS}(C|D), f(x,a) \neq f(y,a), a \in C\}$, where $\mathrm{DIS}(C|D)$ is the relative discernibility relation.

The relative discernibility relation of five typical reducts can be written as follows.

(1) The relative discernibility relation of PRPR is defined by $\mathrm{DIS}(C|D) = \{\langle x,y\rangle \,|\, x,y \in \mathrm{POS}_C(D) \wedge f(x,D) \neq f(y,D) \vee x \in \mathrm{POS}_C(D) \wedge y \notin \mathrm{POS}_C(D)\}$;
(2) The relative discernibility relation of GDPR is defined by $\mathrm{DIS}(C|D) = \{\langle x,y\rangle \,|\, \delta_C(x) \neq \delta_C(y)\}$;
(3) The relative discernibility relation of DPR is defined by $\mathrm{DIS}(C|D) = \{\langle x,y\rangle \,|\, \mu_C(x) \neq \mu_C(y)\}$;
(4) The relative discernibility relation of MDPR is defined by $\mathrm{DIS}(C|D) = \{\langle x,y\rangle \,|\, \phi_C(x) \neq \phi_C(y)\}$;
(5)The relative discernibility relation of DRPR is defined by $\mathrm{DIS}(C|D) = \{\langle x,y\rangle \,|f(x,C) \neq f(y,C) \wedge f(x,D) \neq f(y,D)\}$. Based on the discernibility matrix, one can get the reduct through the following discernibility function. $DF(DM) = \bigwedge\{\bigvee(DM(x,y)) \,|\, \forall x,y \in U, DM(x,y) \neq \varnothing\}$. The expression $\bigwedge\{\bigvee(DM(x,y)))\}$ is the conjunction of all $\bigvee(DM(x,y))$ while $\bigvee(DM(x,y))$ is the disjunction of condition attributes in $DM(x,y)$. The discernibility function can be transformed to a reduced disjunctive form, and each conjunctor of the reduced disjunctive form is a reduct or a superset of a reduct.

3. Granularity space and the quick general reduction algorithm

In this section, we represent five typical reducts in a unified way, and develop two quick general reduction algorithms. In subSection 3.1, to show the reason why we introduce the granularity space, we firstly analyze some promotable points of the existing general reduct definitions from aspects of the reduct definition construction and reduction algorithm designing. After that, we analyze the relationship between the indiscernibility relation of the granularity space and the indiscernibility relation of five typical reducts to perceive the association between granularity space and reducts. Based on that association, we present five typical reducts in the way of granularity space. At the end of this subsection, we show the power and simplicity of the granularity space-based general reduct definition in the comparison to the existing general reduct definitions. In subSection 3.2, to increase the efficiency of reduction algorithms, we extend the positive region approximation to the granularity space and design a new acceleration strategy called granularity approximation, which expands the acceleration domain from the positive region to the universe of decision tables. Based on granularity approximation, we develop two quick general reduction algorithms and present the relationship between the proposed reduction algorithms and the existing reduction algorithms. In subSection 3.3, we show the difference between the proposed algorithms and the related work in Ref [6]. For a better understanding of proposed algorithms, we provide a calculation example of obtaining positive region preservation reduct using proposed algorithms.

3.1. Granularity space: an alternative of general attribute reduct definition

One of the most important parts in rough sets is the answer to what the reduct is. The general reduct definition proposed by Yao et al. [45] has been proven as a remarkable candidate. It's such a powerful definition that there are so many researchers adopting it to accomplish the reduct definition construction and reduction algorithm designing. However, there is still some room for improvement in Yao's definition. Firstly, we could extend the focus on the subset of C to expand the information scale we take into account during data processing. Secondly, we can enhance the flexibility of reduction algorithm designing by abandoning the way of constructing reduct definition, *i.e.*, using evaluation or fitness function to measure the relationship between different attributes, which results in the complexity of theory for that only one more fitness function was proposed before we could propose a new type of attribute reduction. To achieve these improvements, we would propose a new definition of general reduct.

Information granules naturally give rise to hierarchical structures: the same problem or system can be perceived at different levels of specificity (detail) depending on the complexity of the problem, available computing resources, and particular needs to be addressed [39]. In rough sets, information granules can be represented as a partition of the universe. For the convenience of writing, we denote the partition of the universe U as the granularity to emphasize the hierarchical structure of data. Given a non-empty finite set U, a granularity G of U is defined as

$$G = \left\{ X_i \,|\, \bigcup_{X_i \subset G} X_i = U \wedge \left(X_i \cap X_j = \varnothing, i \neq j\right) \right\}.$$

The indiscernibility relation and the discernibility relation of G is defined as follows.

Definition 1. Given a granularity G of a finite non-empty set U, the indiscernibility relation and the discernibility relation of G are defined by

$$\mathrm{IND}(G) = \{\langle x,y\rangle \,|\, \langle x,y\rangle \in X_i \times X_i \wedge X_i \in G\};$$
$$\mathrm{DIS}(G) = \{\langle x,y\rangle \,|\, (\langle x,y\rangle \in X_i \times X_j, i \neq j) \wedge X_i, X_j \in G\}.$$

The indiscernibility relation of the granularity G of U is.

(1) symmetric, *i.e.*, for $\forall x \in U$, we have $\langle x,x\rangle \in \mathrm{IND}(G)$;
(2) transitive, *i.e.*, for $\forall x,y,z \in U$, if $\langle x,y\rangle, \langle y,z\rangle \in \mathrm{IND}(G)$, we have $\langle x,z\rangle \in \mathrm{IND}(G)$;
(3) relexive, *i.e.*, for $\forall x,y \in U$,if $\langle x,y\rangle \in \mathrm{IND}(G)$, we have $\langle y,x\rangle \in \mathrm{IND}(G)$.

On the other side, the discernibility relation of a granularity G of U is relexive, *i.e.*, for $\forall x,y \in U$, if $\langle x,y\rangle \in \mathrm{DIS}(G)$, one can get $\langle y,x\rangle \in \mathrm{DIS}(G)$. Another point needed to be reminded of is that $\mathrm{IND}(G) \cup \mathrm{DIS}(G) = \{\langle x,y\rangle \,|\, \langle x,y\rangle \in U \times U\}$.

Definition 2. Granularity space can be represented in a two-tuple $GS = (G, O)$, where G is a granularity of a finite non-empty set U, O is a set of operators on elements of G. The ancestor granularity space of G is defined by $\mathrm{ANC}(G) = (G, \{\cup\})$; the granularity subspace of G is defined by $\mathrm{SPR}(G) = (G, \{\propto\})$, where $\propto U = \{P, Q \,|\, P \cup Q = U \wedge P \cap Q = \varnothing\}$.

Granularity space is the set of granularity generated from doing several operations specified by O on G. In particular, we specify $G \in (G, \{\propto\})$ and $G \in (G, \{\cup\})$. As a result, we can construct the following lemma.

Lemma 1. *Given a granularity G of a finite non-empty set U, for $\forall G' \in \mathrm{ANC}(G)$, we have $\mathrm{IND}(G') \supseteq \mathrm{IND}(G), \mathrm{DIS}(G') \subseteq \mathrm{DIS}(G)$; for $\forall G' \in \mathrm{SPR}(G)$, we have $\mathrm{IND}(G') \subseteq \mathrm{IND}(G), \mathrm{DIS}(G') \supseteq \mathrm{DIS}(G)$.*

Proof. Taking into account $\mathrm{IND}(G) \cup \mathrm{DIS}(G) = U \times U$, $\mathrm{IND}(G) \cap \mathrm{DIS}(G) = \varnothing$, all we need to do for proving Lemma 1 true is proving true either $\forall G' \in \mathrm{ANC}(G), \mathrm{IND}(G') \supseteq \mathrm{IND}(G)$ or $\forall G' \in \mathrm{SPR}(G), \mathrm{IND}(G') \subseteq \mathrm{IND}(G)$. According to Definition 2, for $\forall G' \in \mathrm{ANC}(G)$, it can be implied that for $\forall E \in G'$, there exists a set $PE \subseteq G$ such that $\bigcup_{P_i \in PE} P_i = E$. As a result, we can get $\varnothing \subseteq \left\{\langle x,y\rangle \,|\, x \in P_i \in PE, y \in \bigcup_{P_j \in PE - P_i} P_j,\ P_i, P_j \in PE, x,y \in E\right\} \Rightarrow \mathrm{DIS}(G') \subseteq \mathrm{DIS}(G)$. That is to say, Lemma 1 is true.

To enhance the understanding of concepts introduced above, here we provide a calculation example of the granularity space.

Example 1. Given $U = \{x_1, x_2, x_3\}, G = \{\{x_1\}, \{x_2, x_3\}\}$, we know $\mathrm{ANC}(G) = (G, \{\cup\}) = \{G, U\}$. let Q denotes as $\{\{x_1\}, \{x_2\}, \{x_3\}\}$, we have $\mathrm{SPR}(G) = (G, \{\propto\}) = \{G, Q\}$, $\mathrm{IND}(U) = U \times U, \mathrm{DIS}(U) = \varnothing; \mathrm{IND}(G) = \{\langle x_1,x_1\rangle, \langle x_2,x_2\rangle, \langle x_3,x_3\rangle, \langle x_2,x_3\rangle, \langle x_3,x_2\rangle\}, \mathrm{DIS}(G) = \{\langle x_1,x_2\rangle, \langle x_2,x_1\rangle, \langle x_1,x_3\rangle, \langle x_3,x_1\rangle\}, \mathrm{IND}(Q) = \{\langle x_1,x_1\rangle, \langle x_2,x_2\rangle, \langle x_3,x_3\rangle\}, \mathrm{DIS}(Q) = \{\langle x_1,x_2\rangle, \langle x_2,x_1\rangle, \langle x_1,x_3\rangle, \langle x_3,x_1\rangle, \langle x_2,x_3\rangle, \langle x_3,x_2\rangle\}$. It is apparent that $\forall G' \in \mathrm{ANC}(G), \mathrm{IND}(G) \subseteq \mathrm{IND}(G')$ *i.e.*, $\mathrm{DIS}(G) \supseteq \mathrm{DIS}(G')$ and $\forall G' \in \mathrm{SPR}(G), \mathrm{DIS}(G) \subseteq \mathrm{DIS}(G')$ *i.e.*, $\mathrm{DIS}(G) \supseteq \mathrm{DIS}(G')$.

Theorem 1. *Given a decision table $DT = (U, C \cup D, V, f)$ and a granularity $G = U/C$, for any type of reduct B, we have $U/B \in \mathrm{ANC}(G)$.*

$U/B \in \mathrm{ANC}(G)$ can be induced from $B \subseteq C$. In other words, for $\forall B \subseteq C$, if $y \in [x]_C$, we have $y \in [x]_B$. Here we summarize the useful conclusions drawn by analyzing the granularity space. The first is that for $\forall B \subseteq C$, we have $U/B \in \mathrm{ANC}(U/C), \mathrm{IND}(B) \supseteq \mathrm{IND}(C)$ and $\mathrm{DIS}(B) \subseteq \mathrm{DIS}(C)$. Reviewing on the discernibility relation of five typical reducts, we can imply that there are some redundant elements in existing discernibility relation. Secondly, $\mathrm{ANC}(U/C)$ contains all granularity induced by $B \subseteq C$, and it indicates the possibility of designing the reduction algorithms by obtaining some special granularity. Here we re-construct the discernibility relation and the indiscernibility relation of five reducts from the perspective of granularity.

Definition 3. Given a decision table $DT = (U, C \cup D, V, f)$, the discernibility relation and the indiscernibility relation of reducts are defined as follows.

$$\mathrm{DIS}(PRPR) = \{\langle x,y\rangle \,|\, y \notin [x]_C \wedge (x,y \in \mathrm{POS}_C(D) \wedge f(x,D) \neq f(y,D) \vee x \notin \mathrm{POS}_C(D) \wedge y \in \mathrm{POS}_C(D))\}$$
$$\mathrm{IND}(PRPR) = \{\langle x,y\rangle \,|\, y \in [x]_C \vee y \notin [x]_C \wedge (x,y \notin \mathrm{POS}_C(D) \vee x,y \in \mathrm{POS}_C(D) \wedge f(x,D) = f(y,D))\}$$
$$\mathrm{DIS}(GDPR) = \{\langle x,y\rangle \,|\, y \notin [x]_C \wedge \delta_C(x) \neq \delta_C(y)\}$$
$$\mathrm{IND}(GDPR) = \{\langle x,y\rangle \,|\, y \in [x]_C \vee y \notin [x]_C \wedge \delta_C(x) = \delta_C(y)\}$$
$$\mathrm{DIS}(DPR) = \{\langle x,y\rangle \,|\, y \notin [x]_C \wedge \mu_C(x) \neq \mu_C(y)\}$$
$$\mathrm{IND}(DPR) = \{\langle x,y\rangle \,|\, y \in [x]_C \vee y \notin [x]_C \wedge \mu_C(x) = \mu_C(y)\}$$
$$\mathrm{DIS}(MDPR) = \{\langle x,y\rangle \,|\, y \notin [x]_C \wedge \phi_C(x) \neq \phi_C(y)\}$$
$$\mathrm{IND}(MDPR) = \{\langle x,y\rangle \,|\, y \in [x]_C \vee y \notin [x]_C \wedge \phi_C(x) = \phi_C(y)\}$$

$$\mathrm{DIS}(DRPR) = \{\langle x,y\rangle \,|\, y \notin [x]_C \wedge (x \in \mathrm{BND}_C(D) \vee y \in \mathrm{BND}_C(D) \vee x,y \in \mathrm{POS}_C(D) \wedge f(x,D) \neq f(y,D))\}$$
$$\mathrm{IND}(DRPR) = \{\langle x,y\rangle \,|\, y \in [x]_C \vee x,y \in \mathrm{POS}_C(D) \wedge f(x,D) = f(y,D)\}$$

Table 1
A decision table for indicating the information scale of granularity space.

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

For the discernibility relation and the indiscernibility relation of five typical reducts, it is worth stressing that for $\forall Red \in \{PRPR, GDPR, DPR, MDPR, DRPR\}, \mathrm{IND}(Red)$ is.

(1) symmetric, *i.e.*, for $\forall x \in U, \langle x,$ we have $x\rangle \in \mathrm{IND}(Red)$,
(2) transitive, *i.e.*, for $\forall x, y, z \in U$, if $\langle x, y\rangle, \langle y, z\rangle \in \mathrm{IND}(Red)$, we have $\langle x, z\rangle \in \mathrm{IND}(Red)$,
(3) relexive, *i.e.*, for $\forall x, y \in U$, if $\langle x, y\rangle \in \mathrm{IND}(Red)$, we have $\langle y, x\rangle \in \mathrm{IND}(Red)$. On the other side, the discernibility relation of *Red*, *i.e.* $\mathrm{DIS}(Red)$, is relexive, which means that for $\forall x, y \in U$, if $\langle x, y\rangle \in \mathrm{DIS}(Red), \langle y, x\rangle$ is an element of $\mathrm{DIS}(Red)$. In addition, we also have $\mathrm{DIS}(Red) \cap \mathrm{IND}(Red) = \varnothing, \mathrm{IND}(Red) \cup \mathrm{DIS}(Red) = U \times U$. As a result, we can imply the following theorem.

Theorem 2. Given a decision table $DT = (U, C \cup D, V, f)$ and $B \subseteq C$, we have

(1) $\mathrm{DIS}(PRPR) \subseteq \mathrm{DIS}(B) \iff \mathrm{POS}_B(D) = \mathrm{POS}_C(D)$;
(2) $\mathrm{DIS}(GDPR) \subseteq \mathrm{DIS}(B) \iff \forall x \in U, \delta_B(x) = \delta_C(x)$;
(3) $\mathrm{DIS}(DPR) \subseteq \mathrm{DIS}(B) \iff \forall x \in U, \mu_B(x) = \mu_C(x)$;
(4) $\mathrm{DIS}(MDPR) \subseteq \mathrm{DIS}(B) \iff \forall x \in U, \phi_B(x) = \phi_C(x)$;
(5) $\mathrm{DIS}(DRPR) \subseteq \mathrm{DIS}(B) \iff \mathrm{IND}(B|D) = \mathrm{IND}(C|D)$.

Proof. According to definitions of related reducts and the discernibility matrix-based reduction algorithm, it is easy to know that items (2), (3), and (4) of Theorem 2 are true. So here we just prove the correctness of items (1) and (5).

(1) **Sufficiency**: We make an assumption that if $\mathrm{DIS}(PRPR) \subseteq \mathrm{DIS}(B)$, there is an object $x \in \mathrm{POS}_C(D)$ satisfying $x \notin \mathrm{POS}_B(D)$. It can be inferred that there exists an object $y \notin [x]_C, f(y, D) \neq f(x, D)$ satisfying $y \in [x]_B$. Thus, we have $\langle x, y\rangle \notin \mathrm{DIS}(B), \langle x, y\rangle \notin \mathrm{DIS}(PRPR)$ and this conflicts with the definition of $\mathrm{DIS}(PRPR)$. As a result, our assumption is not true, and we know $\mathrm{DIS}(PRPR) \subseteq \mathrm{DIS}(B) \Rightarrow \mathrm{POS}_B(D) = \mathrm{POS}_C(D)$.

Necessity: We make an assumption that if $\mathrm{POS}_B(D) = \mathrm{POS}_C(D)$, there exists a pair $\langle x, y\rangle \in \mathrm{DIS}(PRPR)$ such that $\langle x, y\rangle \notin \mathrm{DIS}(B)$. If $x \in \mathrm{POS}_C(D) \wedge y \in \mathrm{POS}_C(D) \wedge f(x, D) \neq f(y, D) \wedge \langle x, y\rangle \notin \mathrm{DIS}(B)$, we know $x, y \notin \mathrm{POS}_B(D)$ and $POS_B(D) \neq POS_C(D)$. That is conflicted with our assumption, *i.e.*, $\mathrm{POS}_B(C) = \mathrm{POS}_C(D)$. If $x \in \mathrm{POS}_C(D) \wedge y \notin \mathrm{POS}_C(D)$, we know $x \in [y]_B$, and that implies $x \notin \mathrm{POS}_B(D)$ and $\mathrm{POS}_C(D) \neq \mathrm{POS}_B(D)$. In summary, if $\mathrm{POS}_B(D) = \mathrm{POS}_C(D)$, then we have $\mathrm{DIS}(PRPR) \subseteq \mathrm{DIS}(B)$.

(5) **Sufficiency**: Assume that if $\mathrm{DIS}(DRPR) \subseteq \mathrm{DIS}(B)$, there exists a pair $\langle x, y\rangle \notin \mathrm{IND}(C|D) \wedge \langle x, y\rangle \in \mathrm{IND}(B|D)$. That means $\exists y \notin [x]_C, y \in [x]_B, f(x, D) \neq f(y, D)$. Then we know $\langle x, y\rangle \in \mathrm{DIS}(DRPR), \langle x, y\rangle \notin \mathrm{DIS}(B)$, and it conflicts with our assumption $\mathrm{DIS}(DRPR) \subseteq \mathrm{DIS}(B)$. Finally, we get $\mathrm{DIS}(DRPR) \subseteq \mathrm{DIS}(B) \Rightarrow \mathrm{IND}(B|D) = \mathrm{IND}(C|D)$.

Necessity: Assume that if $\mathrm{IND}(B|D) = \mathrm{IND}(C|D)$, we have $\exists\langle x, y\rangle \in \mathrm{DIS}(RPR)$ such that $\langle x, y\rangle \notin \mathrm{DIS}(B)$. If $y \notin [x]_C \wedge (x \in \mathrm{BND}_C(D) \vee y \in \mathrm{BND}_C(D))$, we have $\exists p \in [x]_C, \exists q \in [y]_C, f(p, C) \neq f(q, C), f(p, D) \neq f(q, D)$. Because $\langle x, y\rangle \in \mathrm{IND}(B)$, we have $p \in [q]_B$. We get $\langle p, q\rangle \in \mathrm{IND}(B|D)$ and $\langle p, q\rangle \notin \mathrm{IND}(C|D)$. It is conflicted with $\mathrm{IND}(B|D) = \mathrm{IND}(C|D)$. If $x, y \in \mathrm{POS}_C(D) \wedge f(x, d) \neq f(y, d))$, according to $\langle x, y\rangle \in \mathrm{IND}(B)$, it is obvious that $\langle x, y\rangle \in \mathrm{IND}(B|D)$. Noticing $\langle x, y\rangle \notin \mathrm{IND}(C|D)$, we know that our assumption is not true. In summary, $\mathrm{IND}(B|D) = \mathrm{IND}(C|D) \Rightarrow \mathrm{DIS}(DRPR) \subseteq \mathrm{DIS}(B)$.

For the convenience of writing, we specify that *Red* is an element of $\{PRPR, GDPR, DPR, MDPR, DRPR, DRPR\}$ if there is no additional delaration. According to Theorem 1, 2 and Lemma 1, we can construct a unified definition of general attribute reduct using granularity space.

Theorem 3. *Given a decision table $DT = (U, C \cup D, V, f)$ and the target reduct Red, B is a Red of C iff B satisfies the following conditions.*

(1) $U/B \in \mathrm{ANC}(U/C) \cap \mathrm{SPR}(\mathrm{TGran}(Red))$;
(2) $\forall B' \subset B, U/B'$ *does not satisfy condition* (1).

Here $\mathrm{TGran}(Red)$ *stands for* $U/\mathrm{IND}(Red)$.

In fact, item (1) can be written in short as $U/B \in \mathrm{SPR}(\mathrm{TGran}(Red))$ because of $\forall B \subseteq C, U/B \in \mathrm{ANC}(U/C)$. Theorem 3 makes the reduct space more accurate by replacing $\mathrm{ANC}(U/C) \cap \mathrm{SPR}(U)$ with $\mathrm{ANC}(U/C) \cap \mathrm{SPR}(\mathrm{TGran}(Red))$. The core difference of Theorem 3 to the existing general reduct definitions is the attention paid toward the objects' distribution, *i.e.*, granularity space, instead of the relationship between different attribute sets. There are three improvements of the general reduct defined by the objects' distribution. Firstly, the information space we take into consideration is greater, *i.e.*, $|\mathrm{ANC}(U/C)| \geqslant |\{U/B | B \subseteq C\}$. It is obvious that we have $\forall B \subseteq C, U/B \in \mathrm{ANC}(U/C)$ and $\exists G \in \mathrm{ANC}(U/C)$ satisfies $G \neq U/B$. It can be seen from the data in Table 1 that, the information space may vary when different reduct definitions are taken into account, *i.e.*, if we take Yao's definition as the reference of attribute reduction, the information space we take care of can be expressed as $\{U, U/\{a_1\}, U/\{a_2\}, U/\{a_1, a_2\}\}$; if we consider granularity space as the way of reducts representation, the information space should be $\{U, U/\{a_1\}, U/\{a_2\}, U/\{a_1, a_2\}, \{\{x_1, x_3\}, \{x_2\}\}\}$. The expansion of information space makes it possible to design the more efficient reduction algorithms, and GS is a good example. Secondly, general reducts defined by granularity space is independent of heuristic functions. That is to say, there is no need to design new heuristic functions for proposing new type of attribute reduction. All we need for the construction of a new type of attribute reduction is extending or redefining the definition of target granularity space, *i.e.*, the definition of $\mathrm{TGran}(Red) \cap \mathrm{ANC}(U/C)$. When it comes to attribute reduction approaches, it is more flexible in heuristic functions during designing heuristic attribute reduction algorithms based on granularity space because the input of reduction algorithms is definite, *i.e.*, a given granularity space. For example, GS can work well for obtaining five typical reducts with three classical heuristic functions, *i.e.*, dependency degree, consistency, and entropy. For the limitation of the focus of this paper, we do not further explore it here. Thirdly, we can expand the acceleration domain from positive region into the universe and it is helpful to increase the efficiency of reduction algorithms. We would explain it in Section 3.2.

Keeping in mind that $\mathrm{IND}(Red)$ is an equivalence relation, we can construct Algorithm 3.1 for computing $\mathrm{TGran}(Red)$. First step of CTGA is getting the partition U/C because $\mathrm{TGran}(Red)$ is an element of $\mathrm{ANC}(U/C)$. Then, CTGA scans all the combinations of $\langle e1, e2\rangle, e1, e2 \in U/C$, and merges equivalence classes if $\langle x, y\rangle \in \mathrm{IND}(Red) : x \in e1, y \in e2, e1 \neq e2$. After the merging step, the variable TG stores $\mathrm{TGran}(Red)$. The upper bound of time complexity of computing $\mathrm{TGran}(Red)$ using CTGA is $O\left(|U||C \cup D| + |U|^2\right)$. Noticing that the time complexity of merging step is $O\left(|U|^2\right)$, we design key converters, whose generated keys are then hashed by a function implemented in Python's dictionary, to increase the efficiency of merging step.

Algorithm 3.1 Calculating the target granularity algorithm(CTGA)

Input: decision table $DT = (U, C \cup D, V, f)$ and target reduct $Red \in \{PRPR, GDPR, DPR, MDPR, DRPR\}$
Output: Target granularity of relative reduct.
1: Compute $TG = U/C$
2: **for** $ec1 \in TG$ **do**
3: **for** $ec2 \in TG - ec1$ **do**
4: **if** $\langle x, y\rangle \in \mathrm{IND}(Red) : x \in ec1, y \in ec2$ **then**
5: $ec1 := ec1 \cup ec2$
6: $TG := TG - ec2$
7: **end if**
8: **end for**
9: **end for**
10: **return** TG

Definition 4. Given a decision table $(U, C \cup D, V, f)$ and $U/D = \{D_1, D_2, \cdots, D_n\}$, some key converters are defined as

(1) $\mathrm{Key}(x, PRPR) = \begin{cases} i, & \text{if } x \in \mathrm{POS}_C(D) \wedge f(x, D) = D_i \\ n+1, & \text{otherwise} \end{cases}$;
(2) $\mathrm{Key}(x, GDPR) = \delta_C(x)$;
(3) $\mathrm{Key}(x, DPR) = \mu_C(x)$;
(4) $\mathrm{Key}(x, MDPR) = \phi_C(x)$;
(5) $\mathrm{Key}(x, DRPR) = \begin{cases} i, & \text{if } x \in \mathrm{POS}_C(D) \wedge f(x, D) = D_i \\ f(x, C), & \text{otherwise} \end{cases}$

Along this way, we can construct an auxiliary key converter $\mathrm{Key}_a(\mathrm{Key}(x, Red))$, which maps $\mathrm{Key}(x, Red)$ to a set of objects, for storing the target granularity $\mathrm{TGran}(Red)$. The key converters Key and Key_a can be constructed in the time complexity of

$O(|U||C \cup D|)$ and $O(|U|)$. As a result, the upper bound of time complexity of computing TGran(Red) using CTGKC is $O(|U||C \cup D| + 2|U|)$.

Algorithm 3.2 Calculating target granularity using key converters (CTGKC)

Input: decision table $DT = (U, C \cup D, V, f)$ and target reduct $Red \in \{PRPR, GDPR, DPR, MDPR, DRPR\}$
Output: Target granularity of relative reduct in hash function way.
1: Compute U/C
2: **for** $ec \in U/C$
3: $t := \text{Key}(x, Red), \text{Key}_a(t) := \text{Key}_a(t) \cup \{x\}$, where $x \in ec$
4: **end for**
5: **return** Key_a

3.2. Granularity search: an efficient general reduction method

In this part, we focus on the efficient reduction algorithms. Based on the unified representation of five typical reducts, we present a heuristic function called granularity approximation for efficient attribute reduction algorithms. Subsequently, we develop two quick general attribute reduction algorithms.

Definition 5. Given a decision table $DT = (U, C \cup D, V, f)$ and a granularity G of U, the granularity approximation of G in U/B is defined as $\text{GA}(U/B, G) = \bigcup\{[x]_B \mid x \in U, [x]_B \subseteq [x]_G\}$, where $[x]_G$ stands for a set of objects that belong to the same set in G.

If we take G as equivalence classes determined by an attributes set Z, granularity approximation can be written as $\text{POS}_B(Z)$. It is consistent with the uncertainty processing measurement in Pawlak attribute reduction. Obviously, if $\text{GA}(U/B, \text{TGran}(Red)) = U$, we have $U/B \in \text{SPR}(\text{TGran}(Red))$. Thus, the general reduct Definition 3 can be re-written as follows.

Theorem 4. *Given a decision table $DT = (U, C \cup D, V, f)$ and the target granularity* $\text{TGran}(Red)$, *B is a Red of C iff B satisfies*

(1) $\text{GA}(U/B, \text{TGran}(Red)) = U$
(2) $\forall B' \subset B, \text{GA}(U/B', \text{TGran}(Red)) \neq U$.

For the granularity G of U, it is easy to know that $\forall P \subseteq Q \subseteq C, \text{GA}(U/P, G) \subseteq \text{GA}(U/Q, G)$. According to Definition 5, the attribute significance for given attribute sets can be defined as follows.

Definition 6. Given $DT = (U, C \cup D, V, f)$ and target granularity G, the weeded significance of $a \in B \subseteq C$ in B for G is defined as $Sig^-(a, B, G) = |\text{GA}(U/B, G) - \text{GA}(U/B - \{a\}, G)|$; the joined significance of $a \in C - B$ in B for G is defined as $Sig^+(a, B, G) = |\text{GA}(U/B \cup \{a\}, G) - \text{GA}(U/B, G)$, where $|S|$ represents the cardinality of set S.

To increase the computation efficiency of the significance function, we explore the faster approach with the deletion of objects unrelated to the calculation. For the convenience of writing, we use $Sig^+(a, B, G, U)$ to represent the attribute significance, which denotes the value of the significance measure on the universe U. One can prove the following theorem of rank preservation.

Theorem 5. Given a decision table $DT = (U, C \cup D, V, f)$, an attribute set $B \subseteq C$, a granularity G of U and a set $U' = U - \text{GA}(U/B, G)$, for $\forall a, b \in C - B$, if $Sig^+(a, B, G, U) \leqslant Sig^+(b, B, G, U)$, we have $Sig^+(a, B, G, U') \leqslant Sig^+(b, B, G, U')$.

Proof. From the definition of $Sig^+(a, B, G, U)$, we know that its value only depends on the function $\text{GA}(U/B, G)$. Since $U' = U - \text{GA}(U/B, G)$, one can know $\text{GA}(U'/B, G) = \text{GA}(U/B \cup \{a\}, G) - \text{GA}(U/B, G)$. Therefore, we have

$$\frac{Sig^+(a, B, G, U)}{Sig^+(a, B, G, U')} = \frac{|\text{GA}(U/B \cup \{a\}, G, U) - \text{GA}(U/B, G, U)|}{|\text{GA}(U/B \cup \{a\}, G, U') - \text{GA}(U/B, G, U')|} = 1$$

So $Sig^+(a, B, G, U) > Sig^+(b, B, G, U) \iff Sig^+(a, B, G, U') > Sig^+(b, B, G, U')$.

Now we can construct the attribute reduction algorithm based on granularity approximation as Algorithm 3.3, *i.e.*, GS[1].

[1] This algorithm is not related to the granularity search in granularity computing area.

Table 2
The complexity description.

Algorithms	Compute TGran	Compute core	Iteration structure
QGARA-FS	$O(\|U\|\|C\|)$	$O\left(\|C\|^2\|U\|\right)$	$O\left(\sum_{i=1}^{\|C-cr\|}\|U\|(\|cr\|+i)\right)$
QGARA-BS	$O(\|U\|\|C\|)$	/	$O\left(\sum_{i=1}^{\|C\|}\|U\|(\|C\|-i+1)\right)$
GS	$O(\|U\|\|C\|)$	$O\left(\|C\|^2\|U\|\right)$	$O\left(\sum_{i=1}^{\|C-cr\|}\|U_i\|(\|cr\|+i)\right)$
GSV	$O(\|U\|\|C\|)$	/	$O\left(\sum_{i=1}^{\|C\|}\|U_i\|\times i\right)$

Algorithm 3.3 Granularity Search (GS)

Input: decision table $DT=(U,C\cup D,V,f)$ and target reduct Red
Output: A reduct of C
1: Compute the target granularity G according to Algorithm 3.2;
2: Compute $Sig^{-}(a_k,C,G,U), k\leqslant|C|$
3: Put a_k into $core$, where $Sig^{-}(a_k,C,G,U)\rangle 0$
4: $red:=core, i:=0, U_0:=U$
5: **while** $|U_i|>0$
6: Calculate $a_{max}: a_{max}=arg\max_{a\in C-red}Sig^{+}(a,red,G,U_{i+1})$
7: $red:=red\cup\{a_{max}\}$
8: Compute $U_{i+1}:=U_i-\mathrm{GA}(U_i/red,G,U_i)$
9: $i:=i+1$;
10: **end while**
11: **return** red

The reason why we termed Algorithm 3.3 as granularity search is to emphasize the intuition of finding a granularity in the given space. GS consists of four parts: the computation of target granularity, the calculation of core attributes, the selection of attribute with maximal significance, termination judgment. The complexity of computing TGran(Red) is $O(|U||C\cup D|+2|U|)$; the complexity of calculating core attributes is $O\left(|U||C|^2\right)$; the complexity of remaining steps is $O\left(\sum_{i=1}^{|C-cr|}|U_i|(|cr|+i+1)\right)$, where cr denotes the set of core attributes. According to the attribute reduction algorithms in Ref [6,19], the granularity search algorithm can be simplified to further increase the time efficiency by removing the part of computing core attributes and the attribute with maximal significance. For convenience, we denote granularity search without computing core attributes and the attribute with maximal significance as GSV.

Algorithm 3.4 A Granularity Search Variant(GSV)

Input: decision table $DT=(U,C\cup D,V,f)$ and target reduct Red
Output: A reduct of C
1: Compute the target granularity G according to Algorithm 3.2;
2: $red:=\varnothing, i:=0$ and $U_0:=U$
3: **while** $|U_i|>0$ **do**
4: $flag:=False$
5: **for** $a\in C-red$ **do**
6: **if** $Sig^{+}(a,red,G,U_i)>0$ **then**
7: $flag:=True$
8: $red:=red\cup\{a\}$
9: Compute $U_{i+1}:=U_i-\mathrm{GA}(U_i/red,G,U_i)$
10: $i:=i+1$
11: **end if**
12: **end for**
13: **if** $flag$ is False and $|U_i|>0$
14: $red:=red\cup\{a_0\}$, where a_0 is an arbitrary attribute of C
15: **end if**
16: **end while**
17: **return** red

We present the comparison of the upper bound of relevant algorithms time complexity in Table 2, in which "/" denotes that the step does not exist in this algorithm. QGARA-FS and QGARA-BS are general reduction algorithms proposed in Ref [6]. Taking into consideration the reduction results of QGARA-BS and GSV are partly detemined by the attribute order scanned, we divide four algorithms into two groups for comparison. One is GS *vs.* QGARA-FS; another is GSV *vs.* QGARA-BS. The upper bound of time complexity of GS is $O\Big(|U||C| + |U||C|^2 + \sum_{i=1}^{|C-cr|}|U_i|(|cr| + i + 1)\Big)$, where *cr* denotes core attributes. However, the upper bound of time complexity of QGARA-FS is $O\Big(|U||C| + |U||C|^2 + \sum_{i=1}^{|C-cr|}|U|(|cr| + i + 1)\Big)$. Obviously, the time complexity of GS is lower than that of QGARA-FS. Meanwhile, the upper bound of time complexity of GSV is $O\Big(|U||C| + \Big(\sum_{i=1}^{|C|}|U_i| \times i\Big)\Big)$, which is lower than the upper bound of time complexity of QGARA-BS, *i.e.*, $O\Big(|U||C| + \sum_{i=1}^{|C|}|U|(|C| - i + 1)\Big)$. When it comes to the relationship to the existing heuristic functions, reduction algorithms and acceleration mechanisms of the heuristic reduction algorithm, the granularity approximation is an extension of a dependency degree; two proposed algorithms, *i.e.*, GS and GSV, can be taken as the improved edition of algorithms proposed in Ref. [6], and the reason why GS and GSV are more efficient is that the reduct defined from the perspective of granularity; as an acceleration strategy, the granularity approximation can be taken as the extension or the special case of the positive region approximation. Besides, GS and GSV are kin to discernibility matrix-based reduction algorithms, which pay attention to the discernibility relation of reducts instead of the indiscernibility relation of reducts. It is worth trying to apply GS and GSV to address the problems where discernibility matrix-based reduction algorithms have got applied.

3.3. The analysis of granularity space

In this part, the relation and difference between the perspective of granularity and the relative discernibility relation are analyzed. Furthermore, attribute reduction processes related to two perspectives are also compared. In the end, to provide principled intuition of proposed algorithms, we show the process of GS for obtaining the positive region preservation reduct. For the convenience of comparison and the consistency of writing with Ref [6], let $\Delta \in \{PRPR, GDPR, DPR, MDPR, DRPR\}$ denote as the specific type of reduct, general attribute reduct can be defined in the way of relative discernibility relation as follows.

Definition. Given the general decision table $DT_\Delta = (U, C \cup D, V_\Delta, f_\Delta)$, the attribute subset $B \subseteq C$ is a Δ-reduct of C with respect to D, iff it satisfies the following two conditions:

(1) $\mathrm{IND}(B|D_\Delta) = \mathrm{IND}(C|D_\Delta)$
(2) $\forall B' \subset B, U/B'$ does not satisfy condition (1).

D_Δ is an attribute satisfying $U/D_\Delta = \mathrm{TGran}(\Delta)$.

In comparison to $\mathrm{IND}(B|D_\Delta) = \mathrm{IND}(C|D_\Delta), \mathrm{GA}(U/B, \mathrm{TGran}(Red)) = U$ is simpler because of $|U/C| \leqslant |U|$ and $|\mathrm{IND}(C|D)| \leqslant |U|^2$. In addition, the granularity based reduct definition is more intuitive. For the calculation of general reduct, based on relative discernibility relation, $B \subseteq C$ is a superset of Δ-reduct if $W_\Delta(B|D) = W_\Delta(C|D)$, where $W_\Delta(B|D) = |U|^2 - \overline{W_\Delta(D)} - \overline{W_\delta(B)} + \overline{W_\Delta(D \cup B)} = |U|^2 - \sum_{X_i \in U/D}|X_i|^2 - \sum_{X_j \in U/B}|X_j|^2 + \sum_{X_k \in U/B \cup D}|X_k|^2$; based on the granularity space, it is $\mathrm{GA}(U/B, U/D_\Delta) = U$ for the determination of the superset of Δ-reduct, where $\mathrm{GA}(U/B, U/D) = \bigcup\Big\{[x]_B | [x]_B \subseteq [x]_{D_\Delta}\Big\} = \mathrm{POS}_B(D_\Delta)$. Here we provide an example of presenting the difference between two general reduct definitions and reduction process.

Example 2. A calculation example of the positive region preservation attribute reduction. For Table 3, we have $U = \{x_1, x_2, \cdots, x_{11}\}, C = \{a_1, a_2, \cdots, a_6\}, D = \{d\}$, and $U/C = \{X_1, X_2, X_3, X_4, X_5, X_6\} = \{\{x_1\}, \{x_2\}, \{x_3, x_4\}, \{x_5, x_6\}, \{x_7, x_8, x_9\}, \{x_{10}, x_{11}\}\}, U/D = \{P_1, P_2, P_3\} = \{\{x_1, x_3, x_5, x_7\}, \{x_2, x_4, x_6, x_8, x_9, x_{10}\}, \{x_{11}\}\}$.

Assume that we know $\mathrm{TGran}(PRPR) = \{\{x_1\}, \{x_2\}, \{x_3, x_4, \cdots, x_6\}\}$, and we want to express $B = \{a_1, a_2\}$ is a superset of the positive region preservation reduct, it can be written in relative discernibility realtion way, *i.e.*, $\mathrm{IND}(B|D_\Delta) = \{\langle x_1, x_2\rangle, \langle x_2, x_1\rangle, \langle x_1, x_3\rangle, \langle x_3, x_1\rangle, \cdots\} = \mathrm{IND}(C|D_\Delta)$, or in the way of granularity space, *i.e.*, $U/B \in \mathrm{SPR}(\mathrm{TGran}(Red)), U/B = \mathrm{TGran}(Red)$. Let U/P denote as $\mathrm{TGran}(PRPR), U/C, U/P$ and U/B can be drawn as Fig. 1, in which objects contained in a rectangle belong to the same equivalence class. From the relation of $U/P, U/B$, we can easily know B is a positive region preservation reduct. As mentioned above, the general reduct defined by granularity space is simpler and more intuitive than that defined by the relative indiscernibility relation.

To help the understanding of the proposed algorithms, here we show the process of Algorithm 3.3 for obtaining a positive region preservation reduct.

An example of Algorithm 3.2 for computing the target granularity

Considering $X_1 \subset \mathrm{POS}_C(D)$, we have $\forall x \in X_1, \mathrm{Key}(x, PRPR) = f(x_1, d) = 1, \mathrm{Key}_a(1).add(x_1) = \{x_1\}$.
Considering $X_2 \subset \mathrm{POS}_C(D)$, we have $\forall x \in X_2, \mathrm{Key}(x, PRPR) = f(x_2, d) = 2, \mathrm{Key}_a(2).add(x_2) = \{x_2\}$.
Considering $X_3 \subset \mathrm{BND}_C(D)$, we have $\forall x \in X_3, \mathrm{Key}(x, PRPR) = 4, \mathrm{Key}_a(4).add(X_3) = \{x_3, x_4\}$.
Considering $X_4 \subset \mathrm{BND}_C(D)$, we have $\forall x \in X_4, \mathrm{Key}(x, PRPR) = 4, \mathrm{Key}_a(4).add(X_4) = \{x_3, x_4, x_5, x_6\}$.
Considering $X_5 \subset \mathrm{BND}_C(D)$, we have $\forall x \in X_5, \mathrm{Key}(x, PRPR) = 4, \mathrm{Key}_a(4).add(X_5) = \{x_3, x_4, x_5, x_6, x_7, x_8, x_9\}$.

Table 3
A decision table.

U	a_1	a_2	a_3	a_4	a_5	a_6	d
x_1	0	0	0	0	0	0	1
x_2	1	0	0	0	0	0	2
x_3	1	1	0	0	0	0	1
x_4	1	1	0	0	0	0	2
x_5	1	1	0	0	1	1	1
x_6	1	1	0	0	1	1	2
x_7	1	1	0	1	1	1	1
x_8	1	1	0	1	1	1	2
x_9	1	1	0	1	1	1	2
x_{10}	1	1	1	1	1	1	2
x_{11}	1	1	1	1	1	1	3

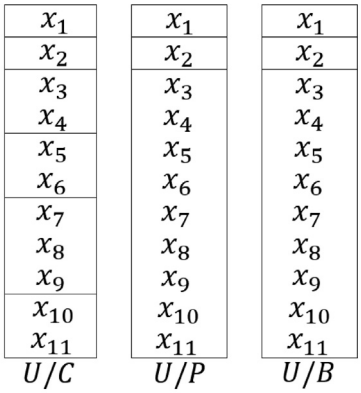

Fig. 1. Granularity comparison.

Considering $X_6 \subset \mathrm{BND}_C(D)$, we have $\forall x \in X_6, \mathrm{Key}(x, PRPR) = 4, \mathrm{Key}_a(4).add(X_6) = \{x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}\}$.

Finally, we get $\mathrm{Key}_a(1) = \{x_1\}, \mathrm{Key}_a(2) = \{x_2\}, \mathrm{Key}_a(4) = \{x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}\}$ and $\mathrm{TGran}(PRPR) = \{\{x_1\}, \{x_2\}, \{x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}\}\}$.

An example for calculating core attributes

According to Algorithm 3.2, we get $G = \mathrm{TGran}(PRPR) = \{\{x_1\}, \{x_2\}, \{x_3, x_4, \cdots, x_{11}\}\}$. Firstly, we compute the partition of U determined by C, *i.e.*, $U/C = \{X_1, X_2, X_3, X_4, X_5, X_6\} = \{\{x_1\}, \{x_2\}, \{x_3, x_4\}, \{x_5, x_6\}, \{x_7, x_8, x_9\}, \{x_{10}, x_{11}\}\}$. Secondly, we calculate the significance of every element of C to determine whether the element is a core attribute or not. For attribute a_1, we have $U/C - \{a_1\} = \{\{x_1, x_2\}, \{x_3, x_4\}, \{x_5, x_6\}, \{x_7, x_8, x_9\}, \{x_{10}, x_{11}\}\}$ and $Sig^-(a_1, C, G) = |U - \{x_3, \cdots, x_{11}\}| = 2$. According to $Sig^-(a_1, C, G) > 0$, we know that a_1 is a core attribute. For attribute a_2, we have $U/C - \{a_2\} = \{\{x_1\}, \{x_2, x_3, x_4\}, \{x_5, x_6\}, \{x_7, x_8, x_9\}, \{x_{10}, x_{11}\}\}$ and $Sig^-(a_2, C, G) = |U - \{x_1, x_5, x_6, \cdots, x_{11}\}| = 3$. According to $Sig^-(a_2, C, G) > 0$, we know that a_2 is a core attribute. For attribute a_3, we have $U/C - \{a_3\} = \{\{x_1\}, \{x_2, x_3, x_4\}, \{x_5, x_6\}, \{x_7, x_8, x_9, x_{10}, x_{11}\}\}$ and $Sig^-(a_3, C, G) = 0$. As a result, we know a_3 is not a core attribute. For attribute a_4, we have $U/C - \{a_4\} = \{\{x_1\}, \{x_2, x_3, x_4\}, \{x_5, x_6, x_7, x_8, x_9\}, \{x_{10}, x_{11}\}\}$ and $Sig^-(a_4, C, G) = 0$. So we know a_4 is not a core attribute. For attributes a_5 and a_6, we have $U/C - \{a_5\} = U/C - \{a_6\} = \{\{x_1\}, \{x_2\}, \{x_3, x_4\}, \{x_5, x_6\}, \{x_7, x_8, x_9\}, \{x_{10}, x_{11}\}\}$ and $Sig^-(a_5, C, G) = Sig^-(a_6, C, G) = 0$. Thus, neither a_5 nor a_6 is not a core attribute. Finally we get $core = \{a_1, a_2\}$.

Example of Algorithm 3.3 for obtaining a positive region preservation reduct

Now we get $G = \mathrm{TGran}(PRPR) = \{\{x_1\}, \{x_2\}, \{x_3, x_4, \cdots, x_6\}\}$ and $reduct = core = \{a_1, a_2\}$. As a result, we can get $U = U - \mathrm{GA}(reduct) = \varnothing$. Because U is $\varnothing$, Algorithm 3.3 outputs $reduct = \{a_1, a_2\}$.

4. Experiments and analyses

The objective of following experiments in this section is to show the effectiveness and the efficiency of the attribute reduction algorithms, *i.e.*, GS and GSV. Experiments are divided into three aspects. First, we employed 12 data sets in Table 4 to verify the performance of time consumption of GS, GSV, QGARA-FS, and QGARA-BS. Then, the computational time of algo-

Table 4
UCI Data Sets.

ID	Data sets	Cases	Attributes	Classes	$\gamma_C(D)$
1	Shuttle	58000	10	7	0.230
2	Mushroom	5644	23	6	0.536
3	Tic	9822	86	2	0.968
4	Segmentation	2310	20	7	0.989
5	Pima-indians-diabetes	768	9	2	0.995
6	Splice	3190	61	3	0.999
7	Dermatology	358	34	6	1
8	Wdbc	569	31	2	1
9	CNAE9	1080	856	9	1
10	Semeion	1593	267	10	1
11	DNA	2000	181	3	1
12	Connect4	67557	43	3	1

rithms GS, GSV, QGARA-FS and QGARA-BS with the increase of the size of objects and attributes was compared. Finally, we evaluated the classification accuracy of reducts generated by general attribute reduction algorithms using the naive bayes classifier and the decision tree classifier.

We carried out all the attribute reduction algorithms in experiments on a personal computer with Windows 10, Intel(R) Core(TM) CPU i5-8265U 1.60GHZ and 8 GB RAM memory. The software used was Visual Studio Code 1.3.8, and the programming language was Python 3.7.

The data sets used in experiments are all downloaded from UCI repository of machine learning data sets [5] whose basic information is outlined in Table 4. For the sake that reduction algorithms can address only symbolic data, data sets containing continuous attributes,*i.e.*, Segmentation, Pima-indians-diabetes, and Wdbc, were preprocessed by equal-width discretization algorithms. For data sets with missing values,*i.e.*, Mushroom, we removed the objects with missing values to achieve uniform treatment of all data sets. The last column of Table 4, *i.e.*, $\gamma_C(D)$, stands for the positive region dependency degree $|POS_C(D)|/|U|$. The data set is consistent when $\gamma_C(D) = 1$; otherwise, the data set is inconsistent. As shown in Table 4, Shuttle, Mushroom, Tic, Segmentation, Pima-indians-diabetes, and Splice are inconsistent and the other 6 data sets are consistent. Taking into consideration the similar results of five types of reducts under the general reduction algorithms, we mainly took positive region preservation reduction and relative discernibility relation preservation reduction results to verify the difference of four reduction algorithms.

4.1. Efficiency comparison of four general attribute reduction algorithms

In this subsection, to show the time efficiency of proposed algorithms, we presented the time consumption of four attribute reduction algorithms in obtaining reducts, and experiments results were shown in Tables 5 and 6.

Table 5 indicated the computational time of QGARA-FS, QGARA-BS, GS, and GSV for obtaining a positive region preservation reduct on 12 data sets. We can see that GSV was the fastest in four attribute reduction algorithms, and GS was faster than QGARA-FS. From results of experiments on both consistent and inconsistent decision tables, the computational time of four algorithms in obtaining positive region preservation reduct followed this order: QGARA-FS $\geqslant$GS, QGARA-FS $>$GSV. GS performed better than QGARA-BS on small-scale data sets. However, in processing the large-scale data, it consumed more time than QGARA-BS for that the computation of the attribute with maximal significance was time-consuming. For most of the cases in experiments, the computational time of GS can reduce over half the computation time of QGARA-FS, such as data sets 2(Mushroom), 3(Tic), 4(Segmentation), etc. In the same condition, GSV can reduce over half of the computation time of QGARA-BS, such as data sets 5(Pima-indians-diabetes), 7(Dermatology), 9(CNAE9), etc. In summary, for calculating positive region preservation reducts on the consistent and inconsistent decision tables, the general attribute reduction algorithms proposed in this paper, *i.e.* GS and GSV, were more efficient than the existing general attribute reduction algorithms, *i.e.*, QGARA-FS and QGARA-BS. Table 6 shows the time consumption of four general reduction algorithms for obtaining a relative discernibility relation preservation reduct. For the consistent decision table, a positive region preservation reduct is also a relative discernibility relation preservation reduct. Thus, the results of the time consumption of four general reduction algorithms on six consistent data sets were similar to the statistics of Table 5. For the time consumption of general reduction algorithms on six inconsistent data sets, we can know that the computational time of GS was less than that of QGARA-FS, and the same condition to GSV and QGARA-BS. In brief, the results of Table 6 were consistent with the observations of Table 5. In summary, for calculating relative discernibility relation preservation reduct on the consistent and inconsistent decision tables, the general reduction algorithms proposed in this paper were more efficient than the existing general reduction algorithms.

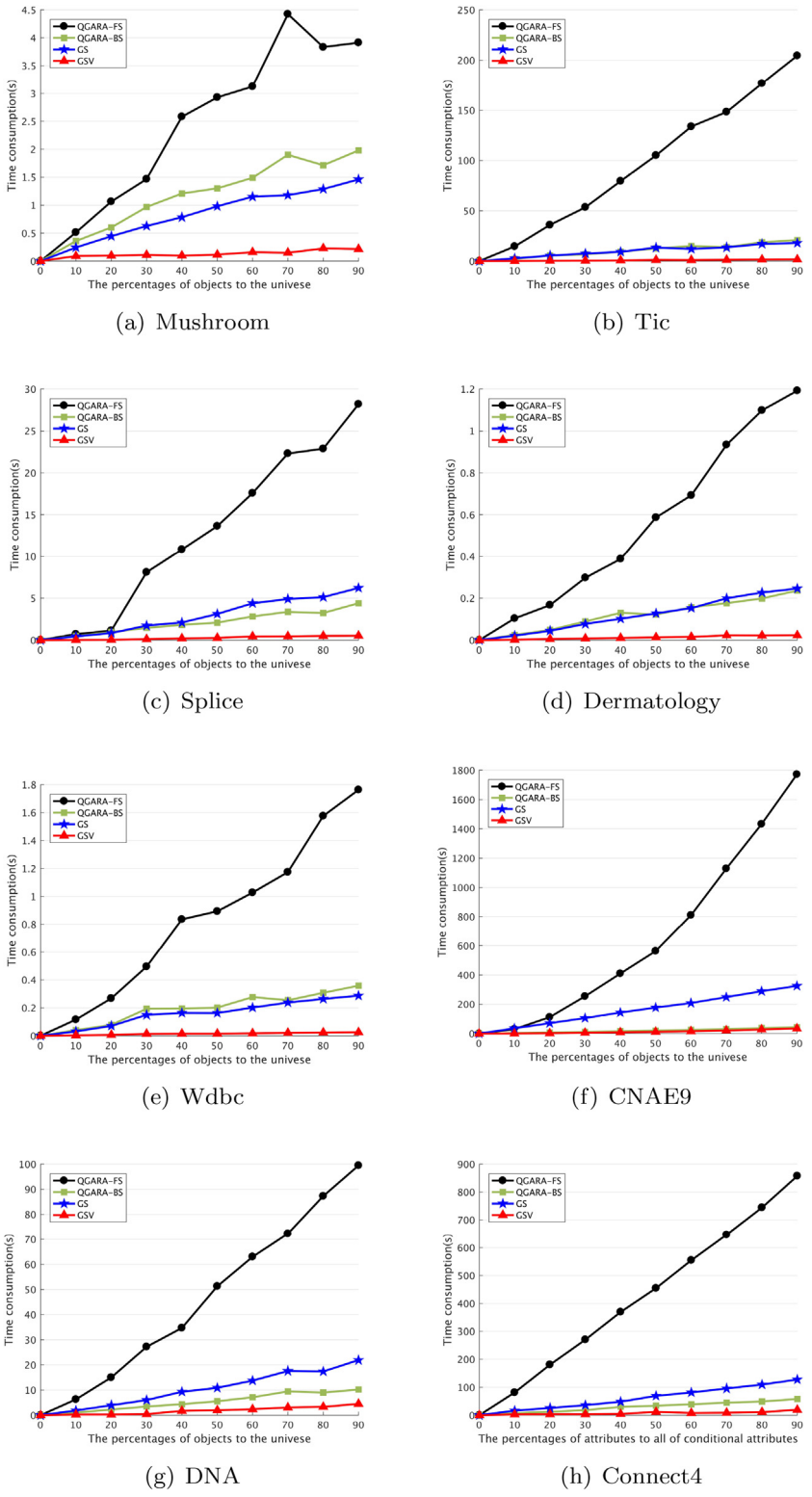

Fig. 2. The time of general reduction algorithms versus the size of objects.

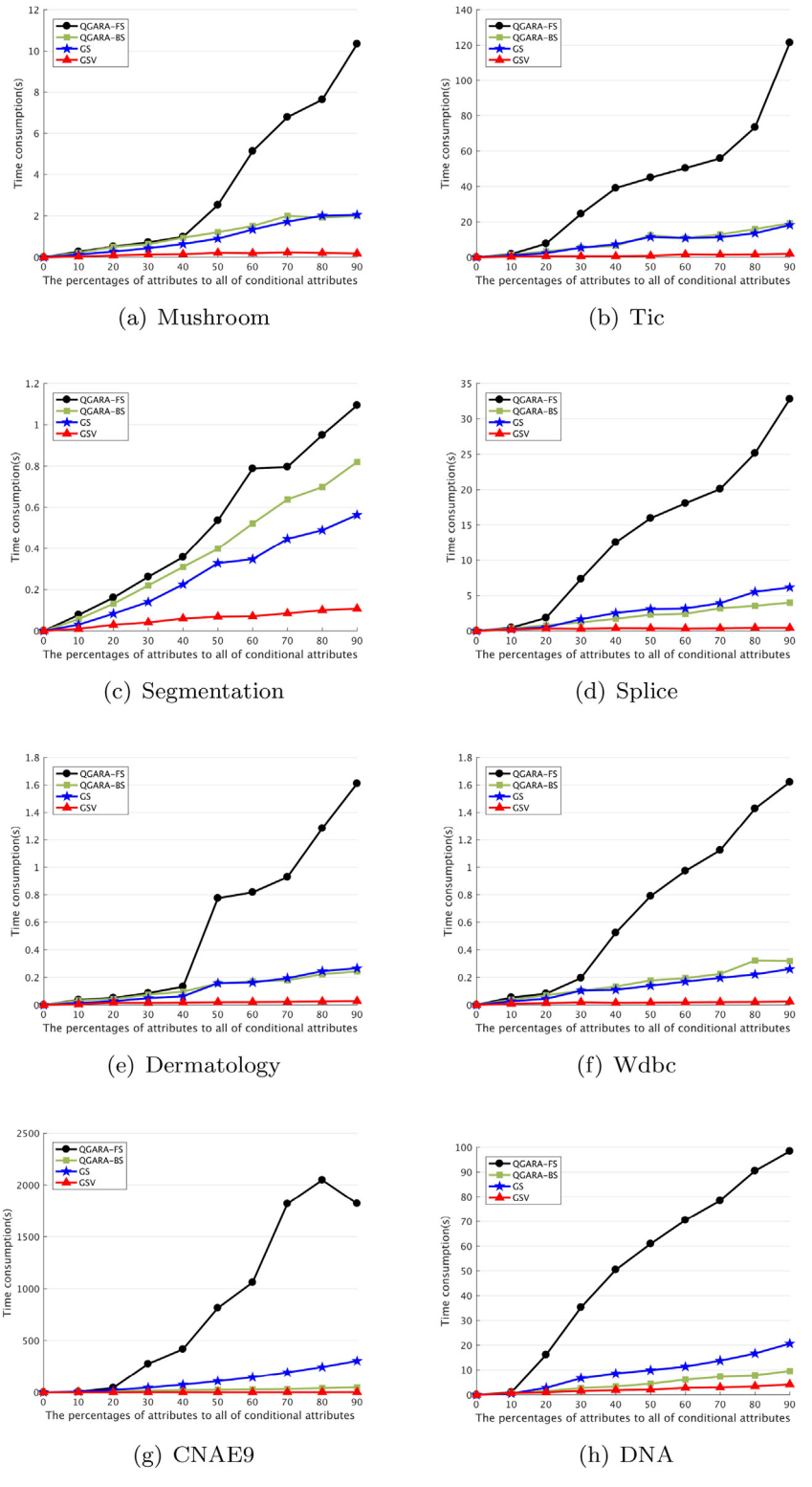

Fig. 3. The time of general reduction algorithms versus the size of attributes.

4.2. Comparison of reduction algorithms for different proportion data sets

In this subsection, to further compare the efficiency of general reduction algorithms, we compared the computational time of QGARA-FS, QGARA-BS, GS, and GSV for obtaining a positive region preservation reduct with the increase of the size of objects and the size of attributes.

Fig. 2 shows detailed change trends of the time consumption of each algorithm for obtaining a positive region preservation reduct with the number of objects increasing. In Fig. 2, the x-coordinate denotes the percentages of objects to the universe, while the y-coordinate concerns the time consumption of algorithms. We employed 8 data sets with different scale (Mushroom, Tic, Splice, Dermatology, Wdbc, CNAE9, DNA, and Connect4) to verify the performance of time consumption of QGARA-FS, QGARA-BS, GS, and GSV. Generally speaking, the computational time of four algorithms increased with the increase of the percentages of objects to the universe. The same as 5 and 6, GS was more efficient than QGARA-FS and GSV was faster than QGARA-BS. When dealing with the same UCI data sets, it is often the case that the computational time of GS was less than that of QGARA-FS, and equal to that of QGARA-BS for small-scale data. But in presence of large-scale data sets, QGARA-BS performed better than GS. It can be observed in many data sets, such as Fig. 2 (c), (e), (g). The computational time of GSV was less than that of the other three general reduction algorithms. The computational time of QGARA-FS increased distinctly in comparison to GS when the number of objects was increasing; the computational time of QGARA-BS increased distinctly in comparison to GSV when the number of objects was increasing.

In Fig. 3, the x-coordinate pertains to the percentages of attributes to the conditional attributes of the data set, while the y-coordinate concerns the time consumption of algorithms. We took 8 data sets (Mushroom, Tic, Segmentation, Splice, Dermatology, Wdbc, CNAE9, and DNA) to verify the performance of the computational time of QGARA-FS, QGARA-BS, GS, and GSV for obtaining a relative discernibility realtion preservation reduct. The result of QGARA-FS, QGARA-BS, GS, and GSV was similar to the result induced from Fig. 2.

Table 5
Time consumption for obtaining PRPR.

ID	Time of QGARA-FS(s)	Time of QGARA-BS(s)	Time of GS(s)	Time of GSV(s)
1	9.326	7.967	6.562	**1.676**
2	11.788	1.990	1.816	**0.213**
3	192.414	18.784	17.890	**1.581**
4	1.215	0.772	0.521	**0.117**
5	0.123	0.109	0.067	**0.018**
6	27.328	3.790	5.638	**0.406**
7	1.139	0.216	0.233	**0.024**
8	1.533	0.272	0.259	**0.040**
9	1095.294	67.414	129.634	**2.199**
10	162.241	13.839	35.478	**2.247**
11	94.026	9.592	21.678	**3.702**
12	889.698	61.010	137.170	**17.500**

Table 6
Time consumption for obtaining DRPR.

ID	Time of QGARA-FS(s)	Time of QGARA-BS(s)	Time of GS(s)	Time of GSV(s)
1	10.885	9.831	7.489	**4.103**
2	3.640	2.419	2.064	**1.864**
3	200.923	19.734	19.231	**3.168**
4	1.423	0.925	0.648	**0.244**
5	0.157	0.140	0.099	**0.044**
6	29.326	4.279	6.016	**0.794**
7	1.145	0.263	0.258	**0.051**
8	1.570	0.357	0.312	**0.065**
9	1152.034	70.652	129.052	**2.061**
10	175.911	14.870	36.988	**2.057**
11	95.743	10.190	20.596	**3.522**
12	894.983	60.119	132.542	**16.472**

Table 7
The classification accuracy of decision tree with PRPR found by five algorithms.

ID	Raw	QGARA-FS	QGARA-BS	GS	GSV	CSFS
2	0.551	0.603	0.608	**0.614**	0.574	0.599
3	0.896	0.892	0.898	0.897	0.896	**0.901**
4	0.943	0.938	0.937	0.937	**0.939**	0.825
5	0.685	0.687	0.689	0.682	0.689	**0.695**
6	0.900	0.656	0.449	0.760	0.804	**0.872**
7	0.936	0.604	0.735	0.789	**0.867**	0.685
8	0.926	**0.940**	0.905	0.916	0.933	0.935
9	0.856	0.873	0.859	0.870	**0.874**	0.867
11	0.901	0.857	0.494	0.871	0.580	**0.932**
12	0.476	0.478	0.462	**0.481**	0.475	0.471
Average	0.807	0.753	0.704	**0.782**	0.763	0.778

Table 8
The classification accuracy of decision tree with DRPR found by five algorithms.

ID	Raw	QGARA-FS	QGARA-BS	GS	GSV	CSFS
2	0.551	0.554	0.551	0.558	0.551	**0.584**
3	0.896	0.896	0.898	0.897	0.894	**0.900**
4	0.943	0.938	0.940	0.938	**0.942**	0.819
5	0.685	0.682	0.685	0.685	**0.691**	0.684
6	0.900	0.651	0.455	0.762	0.809	**0.873**
7	0.936	0.590	0.746	0.787	**0.878**	0.685
8	0.926	**0.938**	0.902	0.907	0.931	0.933
9	0.856	**0.872**	0.853	0.870	0.871	0.862
11	0.901	0.861	0.502	0.868	0.580	**0.933**
12	0.476	**0.477**	0.466	**0.477**	**0.477**	0.472
Average	0.807	0.746	0.700	**0.775**	0.762	**0.775**

Table 9
The classification accuracy of naive bayes with PRPR found by five algorithms.

ID	Raw	QGARA-FS	QGARA-BS	GS	GSV	CSFS
2	0.649	0.557	0.650	0.565	0.582	**0.668**
3	0.777	0.885	0.895	**0.896**	**0.896**	0.805
4	0.603	0.575	0.570	0.575	**0.605**	0.519
5	0.682	**0.682**	**0.682**	**0.682**	**0.682**	**0.682**
6	0.792	0.545	0.519	0.657	0.576	**0.778**
7	0.977	0.602	0.769	0.763	**0.933**	0.682
8	0.902	0.804	0.865	**0.889**	0.874	0.753
9	0.949	0.909	0.901	0.907	**0.933**	0.887
11	0.924	0.767	0.535	0.827	0.594	**0.894**
12	0.597	0.601	0.599	**0.606**	0.597	0.599
Average	0.785	0.693	0.698	**0.737**	0.727	0.727

Table 10
The classification accuracy of naive bayes with DRPR found by five algorithms.

ID	Raw	QGARA-FS	QGARA-BS	GS	GSV	CSFS
2	0.649	0.595	0.642	0.595	0.642	**0.649**
3	0.777	0.891	0.895	**0.896**	**0.896**	0.805
4	0.603	0.575	0.570	0.575	**0.605**	0.519
5	0.682	**0.682**	**0.682**	**0.682**	**0.682**	**0.682**
6	0.792	0.545	0.519	0.657	0.576	**0.778**
7	0.977	0.602	0.769	0.763	**0.933**	0.682
8	0.902	0.804	0.865	**0.889**	0.874	0.753
9	0.949	0.909	0.901	0.907	**0.933**	0.887
11	0.924	0.767	0.535	0.827	0.594	**0.894**
12	0.597	0.601	0.599	**0.606**	0.597	0.599
Average	0.785	0.697	0.698	**0.740**	0.733	0.725

4.3. Comparison of classification accuracy for general attribute reduction algorithms

As we know, there are many factors to the diversity of reducts obtained by reduction algorithms, such as reduction criterion, search strategy, and heuristic functions used, etc. That is to say, the different general reduction algorithms with the same reduction criterion may generate different reducts. To evaluate the effect of reduct obtained by different general reduction algorithms, we randomly selected 10 data sets as test objects from Table 4. We utilized the original data and the reduced data, which is generated by five algorithms QGARA-FS, QGARA-BS, GS, GSV, and chi-square feature selection(CSFS for short), to train naive bayes classifier and decision tree classifier based on the 10-fold cross-validation method. For chi-square feature selection, naive bayes classifier, and decision tree classifier, we used its implementation in [25]. Regarding the parameter K in CSFS, determining how many attributes are contained in reduced data, we assigned K as the cardinality of the reduct generated by GS. It is worth noticing that CSFS is not related to attribute reduction in theory and the reason why we put it into comparisons is to do the evaluation of GS and GSV in feature selection perspective. For convenience of comparison, we take the output of CSFS as a PRPR when K is assigned with the cardinality of the PRPR generated by GS; we take the output of CSFS as a DRPR when K is assigned with the cardinality of the DRPR generated by GS. The classification accuracy to the raw data and the reduced data generated by different algorithms were shown in Tables 7–10, where the column "Raw" represents the classification accuracies of the classifier trained on raw data sets, the boldface highlights the highest accuracy among different algorithms, and the row "Average" represents average classification accuracy of reduction algorithms on 10 data sets, which can be interpreted as an estimated value of classification accuracy obtained by the output of related reduction algorithm over unknown data sets. Obviously, for most of reduced data sets, reduced data can retain similar classification accuracy as the entire data set.

For Table 7, the order of algorithms in the number of achieving the most classification accuracy is CSFS(4) >GSV(3) >GS (2) Q >GARA-FS(1) >QGARA-BS(0). The order of algorithms in the average of classification accuracy on 10 data sets is GS (0.782) >CSFS(0.778) >GSV(0.763) >QGARA-FS(0.753) >QGARA-BS(0.704). GS achieves the best average classification accuracy on 10 data sets. That is to say, the steadiness of algorithms QGARA-FS, QGARA-BS, CSFS, and GSV in classification accuracy is not as good as GS. Furthermore, the PRPR obtained by GS and GSV performs better than that obtained by QGARA-FS and QGARA-BS in the average classification accuracy of decision tree classifier. When it comes to the reduced data generated in the criterion of relative discernibility relation preservation reduction, GSV and CSFS obtain the highest classification accuracy 4 times; QGARA-FS obtains the highest classification accuracy 3 times; GS obtains the highest classification accuracy 1 time; QGARA-BS obtains the highest classification accuracy 0 times. GSV and CSFS perform the best in times of achieving the best classification accuracy. Observing the average classification accuracy on ten data sets, GS is also the best of five, *i.e.*, GS (0.775) $\geqslant$CSFS(0.775) >GSV(0.762) >QGARA-FS(0.746) >QGARA-BS(0.700). As a result, the DRPR obtained by GS and GSV performs better than that obtained by QGARA-FS and QGARA-BS in the average classification accuracy of decision tree classifier.

In the classification accuracy results of naive bayes classifier on reduced data generated in the criterion of positive region preservation reduction and relative discernibility relation preservation reduction, GS was the best one in the classification accuracy average on 10 data sets, and GSV was the second.

Furthermore, we also used the t-test to compare the average 10-fold cross-validation based accuracies over each dataset. Taking the classification accuracies of reducts obtained by GS, GSV, QGARA-FS, and QGARA-BS as the sampling results of four random variables V_{GS}, V_{GSV}, V_{QFS} and V_{QBS}, we set up original hypothesis as $H_0 : V_{GS} = V_{QFS} \vee V_{GSV} = V_{QBS}$ and assigned 0.05 as the significance level α. For convenience of evaluation, we took as a comparison case the average 10-fold cross-validation based accuracies of the same classification algorithm (naive bayes or decision trees) with two reducts (PRPR or DRPR) generated by the comparative algorithms (GS *v.s.* QGARA-FS or GSV *v.s.* QGARA-BS) over a dataset, and there are 80 (2 classification algorithms $\times$ 2 types of reduct $\times$ 2 comparisons $\times$ 10 datasets) comparison cases in experiment. In experimental result, there are only 19 cases rejecting the original hypothesis. That is to say, from the perspective of statistics inference, there are only 23.75% of cases actually supporting that the proposed algorithms are better than the existing algorithms; the remaining 76.25% of cases support the original hypothesis. As a result, four algorithms tie in the aspect of classification accuracy, and if time permitted, it is a reasonable choice of selecting the existing general reduction algorithms for data processing.

In the experimental part, we made a series of comparisons between the proposed general attribute reduction algorithms and the existing general attribute reduction algorithms. We could draw conclusions listed as.

(1) In time consumption of the algorithms to obtain reducts, GS performed well in dealing with small-scale data. When processing large-scale data sets, GSV and QGARA-BS were good choices for attribute reduction;
(2) According to the experiments, the classification accuracies of reducts generated by GS and GSV are competitive as that generated by QGARA-FS and QGARA-BS.

5. Conclusion

In this study, we focus on the effective and efficient general reduction approach to obtain five types of reducts on the complete decision tables. We introduce a concept termed as granularity space and represent five typical reducts with gran-

ularity space. Based on the unified representation, we develop two quick general reduction algorithms. In comparison to the existing general reduction algorithms, the proposed algorithms have two advantages as follows.

(1) The proposed algorithms are more efficient. In the process of attribute reduction, GS can reduce one half of the computation time of QGARA-FS, and GSV can reduce over one half of the computation time of QGARA-BS.
(2) The reducts generated by proposed algorithms perform well as that generated by the existing reduction algorithms in the classification accuracy of decision tree classifier and naive bayes classifier. Over the data sets in experiments, the average classification accuracy of decision tree classifier and naive bayes classifier trained on reduced data generated by GS and GSV is equal to or higher than that trained on reduced data generated by QGARA-FS and QGARA-BS.

However, it is notable that the general reduction definitions and algorithms proposed in this paper are only suitable for the complete decision table. There exist many generalized decision tables, such as incomplete decision tables, interval-valued decision tables. Research on the extension of granularity space for the generalized decision tables will be investigated in future work.

CRediT authorship contribution statement

Baizhen Li: Conceptualization, Methodology, Data curation, Writing - original draft, Software. **Zhihua Wei:** Conceptualization, Supervision. **Duoqian Miao:** Conceptualization, Methodology, Supervision. **Nan Zhang:** Data curation, Visualization, Investigation. **Wen Shen:** Supervision, Validation. **Chang Gong:** Writing - review & editing, Validation. **Hongyun Zhang:** Writing - review & editing, Validation. **Lijun Sun:** Writing - review & editing, Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

The work is partially supported by the National Key Research and Development Project (No. 213), the National Nature Science Foundation of China (No. 61976160, No. 61673301) and Key Lab of Information Network Security, Ministry of Public Security (No. C18608).

References

[1] Ron Tor Das, Kai Keng Ang, Chai Quek, ierspop: a novel incremental rough set-based pseudo outer-product with ensemble learning, Appl. Soft Comput. 46 (2016) 170–186.
[2] Manoranjan Dash, Huan Liu, Consistency-based search in feature selection, Artif. Intell. 151 (1) (2003) 155–176.
[3] Da Yong Deng, Hou Kuan Huang, A new discernibility matrix and function, in: International Conference on Rough Sets and Knowledge Technology, Springer, 2006, pp. 114–121.
[4] Ting Quan Deng, Cheng Dong Yang, Yue Tong Zhang, Xin Xia Wang, An improved ant colony optimization applied to attributes reduction, in: Fuzzy Information and Engineering, Springer, 2009, pp. 1–6.
[5] Dheeru Dua, Casey Graff, UCI machine learning repository, 2017..
[6] Hao Ge, Long Shu Li, Xu. Yi, Chuan Jian Yang, Quick general reduction algorithms for inconsistent decision tables, Int. J. Approximate Reasoning 82 (2017) 56–80.
[7] Hu. Qing Hua, Hui Zhao, Zong Xia Xie, Yu. Da Ren, Consistency based attribute reduction, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2007, pp. 96–107.
[8] Xiao Hua Hu, Nick Cercone, Learning in relational databases: a rough set approach, Computat. Intell. 11 (2) (1995) 323–338.
[9] Hu. YiChung, Flow-based tolerance rough sets for pattern classification, Appl. Soft Comput. 27 (2015) 322–331.
[10] Kuang Yu Huang, An enhanced classification method comprising a genetic algorithm, rough set theory and a modified pbmf-index function, Appl. Soft Comput. 12 (1) (2012) 46–63.
[11] Kuang Yu Huang, I-Hui Li, A multi-attribute decision-making model for the robust classification of multiple inputs and outputs datasets with uncertainty, Appl. Soft Comput. 38 (2016) 176–189.
[12] Richard Jensen, Qiang Shen, Semantics-preserving dimensionality reduction: rough and fuzzy-rough-based approaches, IEEE Trans. Knowled. Data Eng. 16 (12) (2004) 1457–1471.
[13] Xiu Yi Jia, Wen He Liao, Zhen Min Tang, Lin Shang, Minimum cost attribute reduction in decision-theoretic rough set models, Inf. Sci. 219 (2013) 151–167.
[14] Xiu Yi Jia, Lin Shang, Bing Zhou, Yi Yu Yao, Generalized attribute reduct in rough set theory, Knowl.-Based Syst. 91 (2016) 204–218.
[15] Yılmaz Kaya, Murat Uyar, A hybrid decision support system based on rough set and extreme learning machine for diagnosis of hepatitis disease, Appl. Soft Comput. 13 (8) (2013) 3429–3438.
[16] Liang Jun Ke, Zu Ren Feng, Zhi Gang Ren, An efficient ant colony optimization approach to attribute reduction in rough set theory, Pattern Recogn. Lett. 29 (9) (2008) 1351–1357.
[17] Marzena Kryszkiewicz, Comparative study of alternative types of knowledge reduction in inconsistent systems, Int. J. Intell. Syst. 16 (1) (2001) 105–120.
[18] Fei Li, Yun Qiang Yin, Approaches to knowledge reduction of covering decision systems based on information theory, Inf. Sci. 179 (11) (2009) 1694–1704.
[19] Min Li, Chang Xing Shang, Sheng Zhong Feng, Jian Ping Fan, Quick attribute reduction in inconsistent decision tables, Inf. Sci. 254 (2014) 155–180.

[20] Ji Ye Liang, Jun Rong Mi, Wei Wei, Feng Wang, An accelerator for attribute reduction based on perspective of objects and attributes, Knowl.-Based Syst. 44 (2013) 90–100.
[21] Qi He Liu, Leiting Chen, Jian Zhong Zhang, Fan Min, Knowledge reduction in inconsistent decision tables, in: Xue Li, Osmar R. Zaïane, Zhanhuai Li (Eds), Advanced Data Mining and Applications, pages 626–635, Berlin, Heidelberg, 2006. Springer, Berlin Heidelberg..
[22] Zu Qiang Meng, Zhong Zhi 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] Duo Qian Miao, Yan Zhao, Yi Yu Yao, H.X. Li, Fei Fei Xu, Relative reducts in consistent and inconsistent decision tables of the pawlak rough set model, Inf. Sci. 179 (24) (2009) 4140–4150.
[24] Zdzisław Pawlak, Rough sets, Int. J. Comput. Inf. Sci. 11 (5) (Oct 1982) 341–356.
[25] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: machine learning in Python, J. Mach. Learn. Res. 12 (2011) 2825–2830.
[26] Jin Qian, Duo Qian Miao, Z.H. Zhang, W Li, Hybrid approaches to attribute reduction based on indiscernibility and discernibility relation, Int. J. Approximate Reasoning 52 (2) (2011) 212–230.
[27] Yu.Hu.a. Qian, Ji Ye Liang, Witold Pedrycz, Chuang Yin Dang, Positive approximation: an accelerator for attribute reduction in rough set theory, Artif. Intell. 174 (9–10) (2010) 597–618.
[28] Yu. Jiao Shi, Ying Zhou Lei, Maoguo Gong, Enhanced rough–fuzzy c-means algorithm with strict rough sets properties, Appl. Soft Comput. 46 (2016) 827–850.
[29] Andrzej Skowron, Cecylia Rauszer, The discernibility matrices and functions in information systems, in: Intelligent decision support. Springer, 1992, pp. 331–362..
[30] Andrzej Skowron, Jaroslaw Stepaniuk, Towards an approximation theory of discrete problems, part i, Fundam. Inform. 15 (2) (1991) 187–207.
[31] Hai Rong Sun, Wang Rui, Bi Xia Xie, Tian Yao, Continuous attribute reduction method based on an automatic clustering algorithm and decision entropy, in: Control Conference, 2017.
[32] K. Thangavel, A. Pethalakshmi, Dimensionality reduction based on rough set theory: a review, Appl. Soft Comput. 9 (1) (2009) 1–12.
[33] K. Thangavel, A. Pethalakshmi, Dimensionality reduction based on rough set theory: a review, Appl. Soft Comput. 9 (1) (2009) 1–12.
[34] Jing Tian, Quan Wang, Yu. Bing, Yu. Dan, A rough set algorithm for attribute reduction via mutual information and conditional entropy, in: 2013 10th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), IEEE, 2013, pp. 567–571.
[35] Feng Wang, Jiye Liang, Chuangyin Dang, Attribute reduction for dynamic data sets, Appl. Soft Comput. 13 (1) (2013) 676–689.
[36] Jue Wang, Duo Qian Miao, Analysis on attribute reduction strategies of rough set, J. Comput. Sci. Technol. 13 (2) (1998) 189–192.
[37] Jue Wang, Ju. Wang, Reduction algorithms based on discernibility matrix: the ordered attributes method, J. Comput. Sci. Technol. 16 (6) (2001) 489–504.
[38] Ling Wei, Jian Jun Qi, Relation between concept lattice reduction and rough set reduction, Knowl.-Based Syst. 23 (8) (2010) 934–938.
[39] Pedrycz Witold, Granular computing for data analytics: a manifesto of human-centric computing. IEEE/CAA J. Automatica Sinica, 5(6) 1025–1034..
[40] Xiaojun Xie, Xiaolin Qin, Yu. Chunqiang, Xu. Xingye, Test-cost-sensitive rough set based approach for minimum weight vertex cover problem, Appl. Soft Comput. 64 (2018) 423–435.
[41] Xu. Zhang Yan, Li Yu Huang, Bing Ru Yang, Efficient attribute reduction algorithm based on skowron discernibility matrix, in: 2009 International Workshop on Intelligent Systems and Applications, IEEE, 2009, pp. 1–4.
[42] Zhang Yan Xu, Zuo Peng Liu, Bing Ru Yang, Wei Song, A quick attribute reduction algorithm with complexity of max (o (c u), o (c 2 u/c)). Jisuanji Xuebao (Chinese J. Comput.), 29(3) (2006) 391–399..
[43] Tao Yan, Chong Zhao Han, Entropy based attribute reduction approach for incomplete decision table, in: International Conference on Information Fusion, 2017.
[44] Yi Yu Yao, Yan Zhao, Discernibility matrix simplification for constructing attribute reducts, Inform. Sci. 179 (7) (2009) 867–882.
[45] Yiyu Yao, Yan Zhao, Jue Wang, On reduct construction algorithms, in: Transactions on computational science II, Springer, 2008, pp. 100–117.
[46] Yu. Hong, Guo Yin Wang, Fakuan Lan, Solving the attribute reduction problem with ant colony optimization, in: Transactions on rough sets XIII, Springer, 2011, pp. 240–259.
[47] Jianming Zhan, Muhammad Irfan Ali, Nayyar Mehmood, On a novel uncertain soft set model: Z-soft fuzzy rough set model and corresponding decision making methods, Appl. Soft Comput. 56 (2017) 446–457.
[48] Wen Xiu Zhang, Ju Sheng Mi, Wei Zhi Wu, Knowledge reductions in inconsistent information systems, Chinese J. Comput.-Chinese Ed. 26 (1) (2003) 12–18.
[49] Jie Zhou, Duo Qian Miao, Witold Pedrycz, Hong Yun Zhang, Analysis of alternative objective functions for attribute reduction in complete decision tables, Soft. Comput. 15 (8) (2011) 1601–1616.