



Nested structure in parameterized rough reduction



Suyun Zhao^{a,*}, Xizhao Wang^b, Degang Chen^c, Eric C.C. Tsang^d

^a Key Laboratory of Data Engineering and Knowledge Engineering, Renmin University of China, MOE, Room 445, Information Building, No. 59, Zhongguancun Street, Haidian District, Beijing 100872, China

^b School of Mathematics and Computer Sciences, Hebei University, Baoding, China

^c Department of Mathematics and Physics, North China Electric Power University, Beijing, China

^d The Macau University of Science and Technology, Taipa, Macau

ARTICLE INFO

Article history:

Received 3 November 2011

Received in revised form 17 May 2013

Accepted 30 May 2013

Available online 14 June 2013

Keywords:

Parameterized rough sets

Attribute reduction

Variable precision

Nested structure

ABSTRACT

In this paper, by strict mathematical reasoning, we discover the relationship between the parameters and the reducts in parameterized rough reduction. This relationship, named the nested reduction, shows that the reducts act as a nested structure with the monotonically increasing parameter. We present a systematic theoretical framework that provides some basic principles for constructing the nested structure in parameterized rough reduction. Some specific parameterized rough set models in which the nested reduction can be constructed are pointed out by strict mathematical reasoning. Based on the nested reduction, we design several quick algorithms to find a different reduct when one reduct is already given. Here 'different' refers to the reducts obtained on the different parameters. All these algorithms are helpful for quickly finding a proper reduct in the parameterized rough set models. The numerical experiments demonstrate the feasibility and the effectiveness of the nested reduction approach.

© 2013 Elsevier Inc. All rights reserved.

1. Introduction

Rough set (RS) theory, which was proposed by Pawlak [24,25], is a mathematical tool to handle uncertainty of indiscernibility. RS is effective in many real applications such as artificial intelligence, data mining and pattern recognition. However, it is limited by its basic definition 'equivalence relation'. As a result, many generalizations, such as fuzzy rough sets [7,8,22,59,61], cover rough sets [48,49,56], Bayesian rough sets [18,32,36,54] and probabilistic rough sets [42,43,51,55], have been proposed. These generalizations make rough sets feasible to handle many types of practical problems such as the problems with real values, the problems with missing values and the problems with random uncertainty.

One of the important applications of rough set theory is attribute reduction [11,14,34,37,45,47,62,63]. In recent years, researchers, motivated by a desire to acquire reduction robustly, have proposed many methods to mining valuable and less-sensitive attributes by incorporating parameters into rough set theory [15,18,21,57,58,60]. These methods are called parameterized rough reduction. Roughly speaking, parameterized rough reduction is split into two types: reduction on parameterized rough models and parameterized reduction on rough sets. The parameterized rough models share a common characteristic by introducing parameters into the rough approximation operators, whereas parameterized reduction deletes redundant attributes by introducing thresholds into the process of attribute reduction.

* Corresponding author.

E-mail addresses: zsyrose2007@yahoo.com.cn (S. Zhao), xizhaowang@ieee.org (X. Wang), chengdegang@263.net (D. Chen), cctsang@must.edu.mo (E.C.C. Tsang).

Many parameterized rough models are presently being proposed and studied intensively, they are Variable Precision Rough Set Models [2,3,50,53], Robust Fuzzy Rough Set Models [1,9,10,26,45], Probabilistic Rough Set Models [42,43,51], Decision Theoretic Rough Set Models [17,39,40], and Bayesian Rough Set Models [18,32]. The Variable Precision Rough Set Models treat the required parameters as a primitive notion [50]. The interpretation and the process of determination of parameters are based on rather intuitive arguments and left to empirical studies [2,3,16], with the lack of theoretical and systematic studies and justifications on the choices of the threshold parameters. K-Nearest Neighbor Fuzzy Rough Sets provide some methods to determine the required parameters [10], whereas the applications of these models focus mainly on making classification predications. Unlike the aforementioned models, the Bayesian Rough Set Models [18,32] attempt to provide an alternative interpretation of the required parameters. The models are based on Bayes' Rule that expresses the change from the a priori probability to the a posteriori probability [40,41]. In these models, the required parameters can be expressed in terms of probabilities. In addition to variable precision analysis and Bayesian analysis, the parameterized approaches have been applied to the theory of rough sets in some other forms, such as decision-theoretic analysis [39,40] and probabilistic analysis [42,43,51]. The probabilistic rough set models and decision-theoretic rough set models provide a unified and comprehensive framework so that many types of parameterized models can be integrated into a whole [38]. These models provide a systematic method for determining the parameters by using more concrete notions of costs and risks.

Instead of introducing parameters into the approximation operators, parameterized reduction on rough sets introduces parameters into the process of attribute reduction. By relaxing the criteria of attribute reduction, some parameters are introduced in the processing (or notions) of attribute reduction, such as the method of fuzzy reductions (Fuzzy-RED) [5,6] and approximate reductions (Approximate-RED) [28–30]. The common characteristic of these methods is that the parameters are put into the process of attribute reduction. The main difference between these models is that the distinct measures of discernibility power are used in the process of attribute reduction, such as dependency function, information entropy and monotonic measure. The determination of these parameters is usually adopted in an intuitive way.

Whether reduction on parameterized models or parameterized reduction on rough sets is applied, a difficulty with many existing studies on parameterized rough reductions is that no interpretation or procedure for calculating the required threshold exists. In real applications, the setting of parameters affects the results of the reduction significantly, and different thresholds may lead to different reduction results. Some researchers realize the significance of the parameters, and they have paid attention to the parameter analysis and selection. Intuitively, the parameters are given by the experience of an expert in most of the parameterized rough reductions [11,12,52]. Some parameterized rough reduction approaches, based on the assumption that the threshold is in an interval that was determined by the desired level of classification performance, proposed methods for finding an appropriate reduct [2,3]. Other researchers have tried to find a reasonable threshold based on an assessment of the minimum acceptable upper bound of the misclassification error [33]. Essentially, these approaches utilize the principle of the 'extent of classification correctness' to determine an appropriate threshold value under the premise that changing the number of attributes has no effect on the classification results [13]. However, such approaches need to set the classification error beforehand. Instead of setting the parameters intuitively, a systematic method for computing parameters in probabilistic rough sets was provided. Yao et al. introduced risk to probabilistic rough sets and proposed decision theoretic rough sets with Bayesian decision procedures [38,41]. These studies have explicitly noted the parameter effect on the values of approximation operators. Although the aforementioned methods adopt and design different methods of determining and interpreting the parameters, these methods share a common characteristic that all of them seek one optimal or suboptimal value to be the required parameter. These methods omit one possible situation that often occurs in real applications: the required parameter changes frequently. That is, the required parameters are not fixed, and these parameters often change with time or with other conditions in real application.

Some researchers have realized the importance of the parameters and have described various methods for setting parameters. Unfortunately, many researchers are still unaware of the connection between reduction and parameters. The existing methods do not indicate how the parameters affect the performance of reduction. Nobody has mentioned the approaches that can determine the proper reduct based on the given reducts. Discovering the inner relationship between the parameters and the reducts in parameterized rough reduction approaches is now a promising and necessary area of research. In this paper, we propose such a method to quickly find reducts on different parameters in parameterized rough reduction by explicitly showing the connections between the reducts and the parameters.

The remainder of this paper is organized as follows. Section 2 gives some preliminaries, such as rough sets and a parameterized rough reduction. After reviewing many approaches to parameterized rough reduction, we discover the inner connection between the reducts and the parameters in Section 3. By strict mathematical reasoning, we describe the connection, which is called the nested structure. A systematic approach to identify the nested structure is proposed based on a strict mathematical foundation. By using the nested reduction structure, Section 4 proposes some algorithms to find the required reducts quickly. By using numerical experiments, Section 5 clearly demonstrates the effectiveness of these algorithms. Finally, a conclusion is drawn in Section 6.

2. Parameterized rough reduction

In this section, we present some methods to address parameterized rough reduction, that is, reduction in parameterized rough models and parameterized reduction on rough sets.

2.1. Reduction in parameterized rough models

2.1.1. Several parameterized rough models

Many parameterized rough models have been proposed by introducing parameters into rough sets, such as Probabilistic Rough Set Models, Variable Precision Rough Set Models, Decision Theoretic Rough Set Models, and Bayesian Rough Set Models. There are many parameterized rough models, but only some models are reviewed in this subsection because some of them will be analyzed in the following part. Similar results can be obtained for the other parameterized rough reductions by using the theoretical methods proposed in this paper.

VPRS [50]: VPRS has been proposed to handle the problems with misclassification by incorporating a controlled degree of misclassification into knowledge representation of rough set formalism. The concept of set approximations with variable precision is then proposed as $\underline{R}_\beta A = \cup\{[x]_R | c([x]_R, A) \leq \beta, x \in U\}$ and $\overline{R}_\beta A = \cup\{[x]_R | c([x]_R, A) < 1 - \beta, x \in U\}$, where $\forall A \subseteq U$, $\beta \in [0, 0.5)$ and $C(X, Y) = 1 - \text{Card}(X \cap Y) / \text{Card}(X)$, if $\text{Card}(X) > 0$ or $c(X, Y) = 0$ if $\text{Card}(X) = 0$.

VPBRS [31]: The Variable Precision Bayesian Rough Set Model is a parametric Bayesian extension of the rough set model, where the set approximations are defined by using the prior probability as a reference. In VPBRS, the positive region is defined as follows: $\text{POS}^\varepsilon(x) = \cup\{E \in \text{Eq} : P(X|E) \geq 1 - \varepsilon(1 - P(X))\}$, where Eq is the partition space of an equivalence relation. VPBRS combines the advantages of VPRS and Bayesian reasoning approaches in the data mining applications.

VPRS-FRS [26]: VPRS-FRS introduced VPRS into Dubois and Prade's fuzzy rough set framework. The authors in [26] introduced a concept named beta-precision aggregation to fuzzy approximation operators. Set approximations in VPRS-FRS were proposed as $R^*A_\beta(x) = \max(I_{R^*A_\beta}, A(x))$; $R_\beta A(x) = \min(I_{R_\beta A}, A(x))$ with $I_{R^*A_\beta}(x) = \max_{y(y \neq x)} \min(R(x, y), A(y))$; $I_{R_\beta A}(x) = \min_{y(y \neq x)} \max(1 - R(x, y), A(y))$.

VPFRS [19,20]: The authors in [19,20] introduced VPRS into a general FRS framework named VPFRS. We focus here on reviewing the lower approximation of VPFRS because it is very important for attribute reduction. The lower approximation in VPFRS was proposed as follows:

$$\mu_{R_u F}(X_i) \begin{cases} \inf_{x \in S_{i_u}} \vartheta(\mu_{X_i}(x), \mu_F(x)), & \text{if } \exists \alpha_u = \sup\{\alpha \in (0, 1] : e_\alpha(X_i, F) \leq 1 - u\}, \\ 0, & \text{otherwise,} \end{cases} \quad \text{where } S_{i_u} = \text{supp}(X_i \cap X_{i_u}^F).$$

This definition can be equivalently represented in the following way:

$$\mu_{R_u F}(X_i) \begin{cases} \inf \begin{cases} \vartheta(\mu_{X_i}(x), \mu_F(x)) \geq \alpha_u \vartheta(\mu_{X_i}(x), \mu_F(x)), & \text{if } \exists \alpha_u = \sup\{\alpha \in (0, 1] : e_\alpha(X_i, F) \leq 1 - u\}, \\ \mu_{X_i}(x) > 0 \end{cases} \\ 0, & \text{otherwise.} \end{cases}$$

FVPRS [45,46]: FVPRS combines FRS and VPRS with the goal of making FRS a special case of FVPRS. FVPRS is less sensitive to both misclassification and perturbation of data. In FVPRS, fuzzy approximation operators are defined as $\underline{R}_{\alpha, \gamma} A(x) = \inf_{A(u) \leq \alpha} \vartheta(R(x, u), \alpha) \wedge \inf_{A(u) > \alpha} \vartheta(R(x, u), A(u))$ and $\overline{R}_{\alpha, \gamma} A(x) = \sup_{A(u) \geq N(\alpha)} T(R(x, u), N(\alpha)) \vee \sup_{A(u) < N(\alpha)} T(R(x, u), A(u))$, $\forall x \in U$.

VQRS [4]: In VQRS, a fuzzy quantification measure named quantifier was introduced to measure whether one fuzzy set R_y was included in another fuzzy set $A : Q_u \left(\frac{R_y \cap A}{|R_y|} \right)$. Then, the lower approximation was proposed as $R \downarrow_{Q_u} A(y) = Q_u \left(\frac{|R_y \cap A|}{|R_y|} \right)$, where Q is a regularly increasing measure that satisfies the boundary conditions.

Remark 1. One common characteristic of these parameterized models is that one or two parameters are introduced into the approximation operators. One main difference in these parameterized models is that the parameters are put in different positions. For example, in VPRS-FRS and VPFRS, the parameter was put in the position of the aggregation operator; in FVPRS, the parameter was put in the position of the fuzzy decision class.

2.1.2. Reduction in parameterized rough models

Reduction in parameterized rough models means the attribute reduction methods based on the parameterized approximation operators. Essentially, there are two types of attribute reduction methods based on parameterized approximation operators: dependence-function-based approach and discernibility-matrix-based approach. The common criterion of attribute reduction in these approaches is to keep the positive region invariant.

Discernibility-matrix-based approach [45]: The discernibility matrix is designed by keeping the discernibility power invariant. In FVPRS, the discernibility matrix is defined as follows. Suppose $U = \{x_1, x_2, x_3, \dots, x_n\}$; by $M_\alpha(U, R, D)$, we denote an $n \times n$ matrix (c_{ij}^α) called the α -discernibility matrix of $(U, R \cup D)$ such that (M 1) $c_{ij}^\alpha = \{a : T(a(x_i, x_j), \lambda) \leq \alpha\}$, $\lambda = \underline{R}_{\alpha, \gamma}[x_i]_D(x)$ for $D(x_i, x_j) = 0$ and (M 2) $c_{ij}^\alpha = \phi$ for $D(x_i, x_j) = 1$.

By using discernibility matrix, the reduct can be defined as follows. Given a fuzzy decision system $FD = (U, R \cup D)$ and $P \subset R$, P is one reduction of R if and only if P is the minimal subset of R satisfying $P \cap c_{ij} \neq \phi$ for every $c_{ij} \neq \phi$. The algorithms designed by using discernibility matrix are called discernibility-matrix-based approach.

Remark 2. Note that the discernibility-matrix-based approach is sensitive to the parameter in the definition of approximation operators because the computing of the discernibility matrix is dependent on the threshold alpha, whereas alpha is exactly the parameter in the approximation operators.

Dependence-function-based approach [15]: In these approaches, the definition of attribute reduction is as follows. Suppose $FD = (U, R \cup D)$ is a fuzzy decision system. Then, $P \subseteq R$ is called a fuzzy decision reduct if the following statements always hold: (1) $Dep(R) = Dep(P)$; (2) $\forall a \in P, Dep(P - \{a\}) < Dep(P)$, where $Dep(R) = \sum_{x \in U} Pos_R(D)(x)$ (dependence function) and $Pos_R(D) = \cup_{x \in U} R_{x, \cdot}|_D$ (positive region). The algorithms designed by keeping the dependence degree invariant are called Dependence-function-based approach.

Remark 3. Note that the dependence-function-based approach is sensitive to the parameter in parameterized approximation operators because the notions of positive region, dependence function and attribute reduct are defined based on parameterized approximation operators.

2.2. Parameterized reduction on rough sets

Parameterized reduction on rough sets is the second type of parameterized method, which focuses on relaxing the attribute reduction criteria. Some approaches relax the attribute reduction criteria to omit the misclassification and perturbation hidden in the databases. If no confusion arises, we call this type of reduction approach: Parameterized-criteria-of-attribute-reduction approach. The method of fuzzy reductions (Fuzzy-RED) and approximate reductions (Approximate-RED) are the examples of this type of attribute reduction approach.

Approximate-RED [28,30]: The concept of approximate reduction was proposed by keeping the quality measure of certain subsets almost unchanged before and after attribute reduction. The approximate reduction proposed in this way was parameterized because it was free from the harsh criteria of attribute reduction. Let G be a monotonic mapping $G: \rho(A) \rightarrow R$, and G is defined as the quality measure. The α -reduct is defined as follows.

G-Reduct: Given $B \subseteq A$ an α -reduct of generalized information for D in $(U, C \cup D)$ if and only if $|G(B) - G(A)| \leq \alpha$ and for every proper subset $C \subset B$ inequality $|G(C) - G(A)| > \alpha$ holds.

Fuzzy-RED [5,6]: In Fuzzy-RED, after describing a monotonic measure, the authors in [7] proposed the fuzzy decision reduct by keeping this measure almost unchanged. Fuzzy-RED was more general than Approximate-RED because the quality measure was a special case of the monotonic measure.

Fuzzy M-decision reduct: Let M be a monotonic $M: \rho(DS) \rightarrow [0, 1]$ mapping, $B \subseteq A$ and $0 < \alpha \leq 1$. B is called a fuzzy M-decision reduct to degree α if $M(B) \geq \alpha$ and for all $B' \subset B$, $M(B') < \alpha$.

Remark 4. The reducts obtained by Approximate-RED and Fuzzy-RED are sensitive to the parameters used in the process of attribute reduction.

2.3. The algorithms to find reducts on different parameters

A great challenge for many existing studies of parameterized rough reduction is to interpret, determine and analyze the required parameters. In general, there are three ways to determine the parameters. One way is the intuitive method [3,12,45], the second is the assumption method [10,33], and the third is the systematic method [38,40]. Regardless of the method used to determine the parameter, it is hypothesized that only one optimal parameter is required. In the real application, it is difficult to define the optimal parameter. Some reducts on suboptimal parameters are generally needed. Nevertheless, the 'suboptimal' is not unique. A method to find reducts on all of the possible parameters, including a method to find reducts on several different parameters, has not yet been studied. There are two intuitive ways to find reducts on all of the possible parameters.

One intuitive way to find the reducts on all of the possible thresholds is to execute the existing algorithms on all of the possible thresholds. This intuitive method is NP complete. An alternative way is to run the existing algorithms with a fixed step length. This intuitive algorithm can be described as follows.

Intuitive Algorithm 1:

- Step 1: Give the step length and the interval that contains all the parameters.
- Step 2: Run certain existing algorithms on each parameter with the given step length.
- Step 3: Obtain all the reducts.

The intuitive **Algorithm 1** cannot, in fact, find the reducts on all of the possible parameters. This algorithm is merely a heuristic algorithm.

Another intuitive way is the assumption that there exist some boundary values on which the reduct size changes. This intuitive algorithm can be designed as follows.

Intuitive Algorithm 2:

- Step 1: Give all the boundary values.
 Step 2: Run the existing algorithm on all of the boundary values.
 Step 3: Obtain all the reducts on all of the possible parameters.

The second algorithm can find the reducts on all of the possible parameters, whereas this algorithm is designed based on the assumption that all of the boundary values are obtained. Nevertheless, there is no mention of how to find all the boundary values; even the existence of these boundary values has not been verified.

Remark 5. Both of these intuitive algorithms repeat the existing algorithms on the different parameters, but they do not study the connection of the reducts to the different parameters. As a result, these two intuitive algorithms are time-consuming, and many repetitions are conducted.

3. The nested structure in parameterized rough reduction

In this section, the nested reduction, which defines the inner relationship between the parameters and the reducts in parameterized rough reduction, is described. First, some definitions of the nested reduction are proposed that describe the nested structure existing in parameterized rough reduction. Then, some theorems are presented. These theorems provide principles that are needed to construct the nested reduction structure. Furthermore, these theorems show that not all parameterized rough models can be used to construct the nested reduction, only the models satisfying the conditions of these theorems can be used to construct the nested reduction. By mathematical reasoning, several methods of parameterized rough reduction in which the nested structure holds are described.

In this paper, both $\text{Reduct}(\alpha)$ and α -reduct represent the reduct obtained for the parameter α for $0 \leq \alpha < 1$. If no confusion arises, the different reducts mean that the reducts are obtained for two different parameters. The different reducts are called the reducts on different approximations. Then, the reducts on all parameters are called the reducts on all of the possible approximations.

3.1. The nested reduction and weakly nested reduction

In this subsection, some notions of the nested reduction are first proposed. Then, some theorems are proposed that supply the conditions of nested structures. These theorems are targeted at three different types of attribute reduction approaches.

First, we would like to propose several notions that describe several types of nested structures of reductions.

Definition 3.1 (Forward Nested Reduction). In a parameterized model, if the reducts on different approximations satisfy $\forall \text{Reduct}(\beta), \exists \text{Reduct}(\alpha)$ s.t. $\text{Reduct}(\alpha) \subseteq \text{Reduct}(\beta)$, we call this type of nested structure 'Forward Nested Reduction'.

Definition 3.2 (Reverse Nested Reduction). In a parameterized model, if the reducts on different approximations satisfy $\forall \text{Reduct}(\beta), \exists \text{Reduct}(\alpha)$ s.t. $\text{Reduct}(\alpha) \supseteq \text{Reduct}(\beta)$, we call this type of nested structure 'Reverse Nested Reduction'.

Definition 3.3 (Weakly Nested Reduction). In a parameterized model, if the reducts on different approximations satisfy $\forall \text{Reduct}(\beta), \exists P \supseteq \text{Reduct}(\beta)$ contains an α -reduct, we call this type of nested structure 'Forward Weakly Nested Reduction'.

Generally, Forward Nested Reduction and Reverse Nested Reduction are called Nested Reduction. Nested Reduction and Weakly Nested Reduction, which describe the nested structure of reducts in parameterized rough set models, are collectively called the nested structure of reducts.

We next propose several theorems that supply some necessary information to construct the nested structure existing in parameterized rough set models.

Theorem 3.1. Suppose $M_\alpha(U, R, D)$ is an $n \times n$ α -discernibility matrix (c_{ij}^α) , and $\alpha > \beta$. If $M_\alpha(U, R, D) \supseteq M_\beta(U, R, D)$, then the following statements hold:

- (1) With the parameter increasing, the forward nested reduction holds.
- (2) With the parameter decreasing, the weakly nested reduction holds.

where $M_\alpha(U, R, D) \supseteq M_\beta(U, R, D)$ is defined by $c_{ij}^\alpha \supseteq c_{ij}^\beta$ for every $1 \leq i, j \leq n$.

Proof

- (1) We need to prove that $\forall \text{Reduct}(\beta)$, $\exists P \subseteq \text{Reduct}(\beta)$ is an α -reduct. Because $\text{Reduct}(\beta)$ is a β -reduct, $\forall c_{ij} \neq \phi$, $\text{Reduct}(\beta) \cap c_{ij}^\beta \neq \phi$. Because $M_\alpha(U, R, D) \supseteq M_\beta(U, R, D)$, we have $c_{ij}^\alpha \supseteq c_{ij}^\beta$. Thus, $\forall c_{ij} \neq \phi$, $\text{Reduct}(\beta) \cap c_{ij}^\alpha \neq \phi$. $\exists P \subseteq \text{Reduct}(\beta)$, P is the minimal reduct satisfying $\forall c_{ij} \neq \phi$, $P \cap c_{ij} \neq \phi$. According to the definition of attribute reduction, P is an α -reduct.
- (2) We need to prove that $\forall \text{Reduct}(\alpha)$, $\exists P \supseteq \text{Reduct}(\alpha)$ contains a β -reduct. Because $\text{Reduct}(\alpha)$ is a α -reduct, $\forall c_{ij} \neq \phi$, $\text{Reduct}(\alpha) \cap c_{ij} \neq \phi$. Because $M_\alpha(U, R, D) \supseteq M_\beta(U, R, D)$, we have $c_{ij}^\alpha \supseteq c_{ij}^\beta$. Thus, $\exists P \supseteq \text{Reduct}(\alpha)$ satisfies the requirement that $\forall c_{ij} \neq \phi$, $P \cap c_{ij}^\beta \neq \phi$. That is, P contains a β -reduct. \square

Theorem 3.1 is proposed to target the discernibility-matrix-based approach. This theorem shows that if the discernibility matrix is nested, then the reducts are also nested.

Theorem 3.2. Suppose (a) $M: \rho(DS) \rightarrow [0, 1]$, which is the discernibility measure, is monotonic; (b) B , as a α -reduct, satisfies $M(B) \geq \alpha$ and $\forall B' \subset B$, $M(B') < \alpha$, then the following statements hold:

- (1) With the parameter increasing, the weakly nested reduction holds.
- (2) With the parameter decreasing, the forward nested reduction holds.

Proof

- (1) $M(\bullet)$ is monotonic $\Rightarrow \exists P \supseteq \text{Reduct}(\beta)$ satisfies $M(P) \geq \alpha \Rightarrow$ by the condition (b) $\forall \text{Reduct}(\beta)$, $\exists P \supseteq \text{Reduct}(\beta)$ contains an α -reduct.
- (2) $M(\bullet)$ is monotonic $\Rightarrow \exists P \subseteq \text{Reduct}(\alpha)$ satisfies $M(P) \geq \beta$ and $\forall P' \subset P$, $M(P') < \beta \Rightarrow P = \text{Reduct}(\beta)$ by the condition (b) $\forall \text{Reduct}(\alpha)$, $\exists P \subseteq \text{Reduct}(\alpha)$ is a β -reduct. \square

Theorem 3.2 is proposed to target the flexible-criteria-based approach, which shows that the nested structure of reducts in the flexible-criteria-based approach depends on the definition of attribute reduction.

Theorem 3.3 is proposed to target the dependence-function-based approach. Comparatively, the result of **Theorem 3.3** is weaker than the results of **Theorems 3.1** and **3.2** because only the weakly nested reduction holds in the results of **Theorem 3.3**.

Theorem 3.3. Suppose $M_\alpha(B)$ is monotonic with B . Then, the following statements hold:

- (1) If $M_\alpha(B)$ is monotonic with α , with the parameter decreasing, the weakly nested reduction holds.
- (2) If $M_\alpha(B)$ is ANTI-monotonic with α , with the parameter increasing, the weakly nested reduction holds.

Proof

- (1) Let $Q = \text{Reduct}(\alpha)$. According to the definition of attribute reduction in the dependence-function-based approach, we have $M_\alpha(Q) = M_\alpha(R)$ and for all $Q' \subset Q$, $M_\alpha(Q') < M_\alpha(R)$. $M_\beta(Q) \leq M_\beta(R) \Rightarrow$ (a) When $M_\beta(Q) = M_\beta(R)$, $Q = \text{Reduct}(\alpha)$ is a β -reduct; (b) when $M_\beta(Q) < M_\beta(R)$, there exists $Q \subset P \subseteq R$ and P is a super β -reduct. $\Rightarrow \forall \text{Reduct}(\alpha)$, $\exists P \supseteq \text{Reduct}(\alpha)$ contains a β -reduct.
- (2) The proof is similar to (1). \square

In this section, we present three theorems. These theorems give some necessary conditions to construct the nested reductions or the weakly nested reductions. These theoretical results supply some principles to judge which parameterized model is suitable to construct a nested reduction. Furthermore, these theoretical results are important and useful in designing algorithms to find reducts on different approximations, or even on all levels of approximations, quickly.

3.2. Some parameterized methods satisfying the nested reduction or the weakly nested reduction

In this subsection, we present some properties of the parameterized models. By these properties and aforementioned theorems, we find some parameterized models in which the nested reduction structures hold.

3.2.1. The discernibility-matrix-based parameterized models

We discuss the discernibility-matrix-based model: fuzzy variable precision rough set model (FVPRS) in this part. Because Theorem 3.1 is proposed for those parameterized models that find reducts based on discernibility matrix, we need to determine whether FVPRS satisfies the condition in Theorem 3.1. First, several properties of FVPRS are described.

Proposition 3.1. In FVPRS, if $\alpha > \beta$, then $R_{\vartheta_\alpha}A(x) \geq R_{\vartheta_\beta}A(x), \forall x \in U$.

Proof. According to the definition of lower approximation in FVPRS, we have

$$\begin{aligned} R_{\vartheta_\alpha}A(x) &= \boxed{\inf_{A(u) \leq \beta} \vartheta(R(x,u), \alpha)} \wedge \boxed{\inf_{\beta < A(u) \leq \alpha} \vartheta(R(x,u), \alpha)} \wedge \boxed{\inf_{A(u) > \alpha} \vartheta(R(x,u), A(u))} \\ R_{\vartheta_\beta}A(x) &= \boxed{\inf_{A(u) \leq \beta} \vartheta(R(x,u), \beta)} \wedge \boxed{\inf_{\beta < A(u) \leq \alpha} \vartheta(R(x,u), A(u))} \wedge \boxed{\inf_{A(u) > \alpha} \vartheta(R(x,u), A(u))} \end{aligned}$$

Because $\inf_{A(u) \leq \beta} \vartheta(R(x,u), \alpha) \geq \inf_{A(u) \leq \beta} \vartheta(R(x,u), \beta)$, $\inf_{\beta < A(u) \leq \alpha} \vartheta(R(x,u), \alpha) \geq \inf_{\beta < A(u) \leq \alpha} \vartheta(R(x,u), A(u))$ and $\inf_{A(u) > \alpha} \vartheta(R(x,u), A(u)) = \inf_{A(u) > \alpha} \vartheta(R(x,u), A(u))$, we have $R_{\vartheta_\alpha}A(x) \geq R_{\vartheta_\beta}A(x), \forall x \in U. \square$

Proposition 3.2 shows how the threshold affects the discernibility power of the attributes.

Proposition 3.2. In FVPRS, if residuated implication ϑ satisfies 'if $\vartheta(a, b) = c$, then $T(a, c) = b, \forall a, b, c \in [0, 1]$ ', then the following statement holds:

If $T(a(x, u), R_{\vartheta_\beta}A(x)) \leq \beta$, then $T(a(x, u), R_{\vartheta_\alpha}A(x)) \leq \alpha$, where $a(x, u)$ is the similarity degree of $x, u \in U$ on the attribute $a \in R$.

Proof. Because

$$R_{\vartheta_\beta}A(x) = \boxed{\inf_{A(u) \leq \beta} \vartheta(R(x,u), \beta)} \wedge \boxed{\inf_{\beta < A(u) \leq \alpha} \vartheta(R(x,u), A(u))} \wedge \boxed{\inf_{A(u) > \alpha} \vartheta(R(x,u), A(u))} \begin{matrix} \text{(a)} & \text{(b)} & \text{(c)} \end{matrix}$$

we have that $R_{\vartheta_\beta}A(x)$ get its value in the following three situations:

- (1) Assume $R_{\vartheta_\beta}A(x)$ has the value determined in part (c). $\exists u' \in U$ s.t. $A(u') > \alpha$ and $R_{\vartheta_\beta}A(x) = \vartheta(R(x, u'), A(u'))$, $A(u') \Rightarrow \vartheta(R(x, u'), A(u'))$ is the minimum value among $\vartheta(R(x, u), \alpha)$ for $A(u) \leq \alpha$ and $\vartheta(R(x, u), A(u))$ for $A(u) > \alpha \Rightarrow R_{\vartheta_\alpha}A(x) = \vartheta(R(x, u'), A(u')) = R_{\vartheta_\beta}A(x) \Rightarrow T(a(x, y), R_{\vartheta_\alpha}A(x)) = T(a(x, y), R_{\vartheta_\beta}A(x)) \leq \beta < \alpha$.
- (2) Assume $R_{\vartheta_\beta}A(x)$ has the value determined in part (b). $\exists u' \in U$ s.t. $\beta < A(u') \leq \alpha$ and $R_{\vartheta_\beta}A(x) = \vartheta(R(x, u'), A(u')) \Rightarrow T(R(x, u'), R_{\vartheta_\beta}A(x)) = A(u')$ (by the given condition) $\Rightarrow a(x, y) \leq R(x, u')$ (by the monotonicity of T-norm) because $T(a(x, y), R_{\vartheta_\beta}A(x)) \leq \beta$ and $\beta < A(u') \leq \alpha$.

There are two possibilities for determining the value of $R_{\vartheta_\alpha}A(x)$: $R_{\vartheta_\alpha}A(x) = \vartheta(R(x, u''), \alpha)$ (here $A(u'') \leq \alpha$) and $R_{\vartheta_\alpha}A(x) = \vartheta(R(x, u''), A(u''))$ (here $A(u'') > \alpha$). $R_{\vartheta_\alpha}A(x) = \vartheta(R(x, u''), \alpha)$ (here $A(u'') \leq \alpha \Rightarrow T(R(x, u''), R_{\vartheta_\alpha}A(x)) \leq \alpha$ (By definition of residuated implication) $\Rightarrow \vartheta(R(x, u''), \alpha) \leq \vartheta(R(x, u'), A(u')) \leq \vartheta(R(x, u'), \alpha) \Rightarrow R(x, u') \leq R(x, u'')$ (by the monotonicity of residuated implication) $\Rightarrow a(x, y) \leq R(x, u'')$ (because $a(x, y) < R(x, u') \Rightarrow T(a(x, y), R_{\vartheta_\alpha}A(x)) \leq \alpha$ (by the monotonicity of T-norm).

$$\begin{aligned} R_{\vartheta_\alpha}A(x) &= \vartheta(R(x, u''), A(u'')) \text{ (here } A(u'') > \alpha) \Rightarrow \vartheta(R(x, u''), A(u'')) \leq \vartheta(R(x, u'), \alpha) \Rightarrow T(R(x, u'), R_{\vartheta_\alpha}A(x)) \\ &= T(R(x, u'), \vartheta(R(x, u''), \alpha)) \leq \alpha \end{aligned}$$

(By the monotonicity of T-norm) because $T(R(x, u'), \vartheta(R(x, u'), \alpha)) \leq \alpha \Rightarrow T(a(x, y), R_{\vartheta_\alpha}A(x)) \leq \alpha$ (because $a(x, y) < R(x, u')$).

- (3) Assume $R_{\vartheta_\beta}A(x)$ has the value determined in part (a). $\exists u' \in U$ s.t. $A(u') \leq \beta$ and $R_{\vartheta_\beta}A(x) = \vartheta(R(x, u'), \beta) \Rightarrow T(R(x, u'), R_{\vartheta_\beta}A(x)) = \beta$ (by the given condition) $\Rightarrow a(x, y) \leq R(x, u')$ (the monotonicity of T-norm) because $T(a(x, y), R_{\vartheta_\beta}A(x)) \leq \beta$. Like (2), we determine that $T(a(x, y), R_{\vartheta_\alpha}A(x)) \leq \alpha$.

All these show that if $T(a(x, u), R_{\vartheta_\beta}A(x)) \leq \beta$ and $\alpha > \beta$, then $T(a(x, u), R_{\vartheta_\alpha}A(x)) \leq \alpha$ in the condition that if $\vartheta(a, b) = c$, then $T(a, c) = b, \forall a, b, c \in [0, 1]. \square$

These results show that if the attribute a is selected as a discerning member into discerning the object pair x and u on a smaller threshold β , then it is also selected as a discerning member on a larger threshold α . That is, some more attributes are selected as discerning members in FVPRS on the larger threshold.

Proposition 3.3. If $\alpha > \beta$, then $M_\alpha(U, R, D) \supseteq M_\beta(U, R, D)$ in FVPRS.

Proof. By definition of α -discernibility matrix in FVPRS and Proposition 3.2, we have that if $a \in c_{ij}^\beta$, then $a \in c_{ij}^\alpha$, i.e., $c_{ij}^\alpha \supseteq c_{ij}^\beta$. Thus, $M_\alpha(U, R, D) \supseteq M_\beta(U, R, D)$ in FVPRS. \square

Proposition 3.3 shows that if the residuated implication satisfies ‘if $\vartheta(a,b) = c$, then $T(a,c) = b$, $\forall a,b,c \in [0,1]$ ’, then the forward nested reduction and the weakly nested reduction can be constructed in FVPRS.

3.2.2. The dependence-function-based models

Some parameterized models, such as Approximate-RED and fuzzy-RED, satisfy the conditions in **Theorem 3.2**. The following proposition shows this fact.

Proposition 3.4. Suppose B is a α -reduct and $M:\rho(DS) \rightarrow [0,1]$ is the discernibility measure,

- (1) In Approximate-RED, (a) $M(\bullet)$ is a monotonic; (b) $M(B) \geq \alpha$; (c) $\forall B' \subset B, M(B') < \alpha$ holds.
- (2) In fuzzy-RED, (a) $M(\bullet)$ is a monotonic; (b) $M(B) \geq \alpha$; (c) $\forall B' \subset B, M(B') < \alpha$ holds.

Proof

- (1) By definition of attribute reduction in [23], $B \subseteq A \Rightarrow |G(B) - G(A)| \leq \alpha \Rightarrow G(B) \geq G(A) - \alpha$ because $G(\cdot)$ is a monotonic \Rightarrow let $\beta = G(A) - \alpha$, then $G(B) \geq \beta$. For every proper subset $C \subset B$, the inequality $|G(C) - G(A)| > \alpha$ holds $\Rightarrow G(C) < G(A) - \alpha = \beta$.
- (2) It is straightforward to get that in fuzzy-RED, (a) $M(\bullet)$ is a monotonic; (b) $M(B) \geq \alpha$; (c) $\forall B' \subset B, M(B') < \alpha$ holds. \square

Proposition 3.4 shows clearly that approximate-RED and fuzzy-RED satisfy the conditions in **Theorem 3.2**. That is, they can construct the weakly nested reduction and the forward nested reduction.

3.2.3. The dependence-function-based models

In this subsection, we present some properties of the dependence-function-based models. These properties show that most of the dependence-function-based models can be used to construct a reverse weakly nested reduction.

Proposition 3.5. In VPBRS, $M_\alpha(B)$ is monotonic with B , and it is monotonic with α .

Proof. By definition of the lower approximation operator in VPBRS,

$$B \subseteq P \Rightarrow E_B \supseteq E_P \Rightarrow p(X/E_B) \leq p(X/E_P) \Rightarrow \underline{Pos}_{B,\alpha}(x) \leq \underline{Pos}_{P,\alpha}(x) \Rightarrow M_\alpha(B) \leq M_\alpha(P).$$

$$\alpha > \beta \Rightarrow 1 - \alpha(1 - p(X)) < 1 - \beta(1 - p(X)) \Rightarrow \underline{Pos}_{B,\alpha}(X) \supseteq \underline{Pos}_{B,\beta}(X) \Rightarrow M_\alpha(B) \leq M_\beta(B). \quad \square$$

Proposition 3.6. In VPFRS, $M_\alpha(B)$ is monotonic with B , and $M_\alpha(B)$ is monotonic with α .

Proof. By definition of the lower approximation operator in VPFRS,

$$B \subseteq P \Rightarrow \forall x \in U, BX_i(x) \geq PX_i(x) \Rightarrow \vartheta(\mu_{BX_i}(x), \mu_F(x)) \leq \vartheta(\mu_{PX_i}(x), \mu_F(x)) \Rightarrow \underline{B}_\alpha(x) \leq \underline{P}_\alpha(x) \Rightarrow \underline{Pos}_{B,\alpha}(x) \leq \underline{Pos}_{P,\alpha}(x) \Rightarrow M_\alpha(B) \leq M_\alpha(P).$$

$$\alpha > \beta \Rightarrow 1 - \alpha < 1 - \beta \Rightarrow e_\alpha(A, B) \geq e_\beta(A, B) \left(\text{here } e_\alpha(A, B) = 1 - \frac{|A \cap \vartheta_\alpha(A, B)|}{|A|} \right)$$

and

$$\{\gamma \in (0, 1] : e_\gamma(X_i, F) \leq 1 - \alpha\} \subseteq \{\gamma \in (0, 1] : e_\gamma(X_i, F) \leq 1 - \beta\} \Rightarrow \underline{B}_\alpha(x) \leq \underline{B}_\beta(x) \Rightarrow \underline{Pos}_{B,\alpha}(x) \leq \underline{Pos}_{B,\beta}(x) \Rightarrow M_\alpha(B) \leq M_\beta(B). \quad \square$$

Proposition 3.7. In VPRS-FRS, $M_\alpha(B)$ is monotonic with B , and ANTI-monotonic with α .

Proof. By definition of the lower approximation operator in VPRS-FRS,

$$\begin{aligned}
B \subseteq P &\Rightarrow \forall x, y \in U, B(x, y) \geq P(x, y) \Rightarrow \max(1 - B(x, y), A(y)) \leq \max(1 - P(x, y), A(y)) \Rightarrow \underline{I}_{B, A_\alpha}(x) \leq \underline{I}_{P, A_\alpha}(x) \\
&\Rightarrow \underline{B}_\alpha(x) \leq \underline{P}_\alpha(x) \Rightarrow \underline{Pos}_{B, \alpha}(x) \leq \underline{Pos}_{P, \alpha}(x) \Rightarrow M_\alpha(B) \leq M_\alpha(P). \\
\alpha > \beta &\Rightarrow \min_\alpha(U) \geq \min_\beta(U) \Rightarrow \underline{I}_{B, A_\alpha}(x) \leq \underline{I}_{B, A_\beta}(x) \Rightarrow \underline{B}_\alpha(x) \geq \underline{B}_\beta(x) \Rightarrow \underline{Pos}_{B, \alpha}(x) \geq \underline{Pos}_{B, \beta}(x) \Rightarrow M_\alpha(B) \leq M_\beta(B). \quad \square
\end{aligned}$$

Proposition 3.8. In VPRS, $M_\alpha(B)$ is monotonic with B , and ANTI is monotonic with α .

Proof. By definition of the lower approximation operator in VPRS,

$$\begin{aligned}
B \subseteq P &\Rightarrow \forall x \in U, [x]_B \supseteq [x]_P \Rightarrow c([x]_B, A) \geq c([x]_P, A) \Rightarrow \cup\{[x]_R | c([x]_B, A) \leq \alpha, x \in U\} \subseteq \cup\{[x]_R | c([x]_P, A) \leq \alpha, x \in U\} \\
&\Rightarrow \underline{B}_\alpha(x) \leq \underline{P}_\alpha(x) \Rightarrow \underline{Pos}_{B, \alpha}(x) \leq \underline{Pos}_{P, \alpha}(x) \Rightarrow M_\alpha(B) \leq M_\alpha(P). \\
\alpha > \beta &\Rightarrow \cup\{[x]_R | c([x]_B, A) \leq \alpha, x \in U\} \supseteq \cup\{[x]_R | c([x]_B, A) \leq \beta, x \in U\} \Rightarrow \underline{B}_\alpha(x) \geq \underline{B}_\beta(x) \Rightarrow \underline{Pos}_{B, \alpha}(x) \geq \underline{Pos}_{B, \beta}(x) \\
&\Rightarrow M_\alpha(B) \leq M_\beta(B). \quad \square
\end{aligned}$$

Proposition 3.9. In VQRS, $\underline{B}_\alpha A(x)$ and $\underline{Pos}_{B, \alpha}(x)$ NOT is monotonic with B .

Proof. By definition of the lower approximation operator in VQRS, $B \subseteq P \Rightarrow \forall y \in U, B_y \supseteq P_y \Rightarrow$ the inequality $\frac{|B_y \cap A|}{|B_y|} \leq \frac{|P_y \cap A|}{|P_y|}$ does not always hold $\Rightarrow \underline{B}_\alpha A(x)$ and $\underline{Pos}_{B, \alpha}(x)$ is not monotonic with B . As a result, $M_\alpha(B)$ is monotonic with B . \square

Propositions 3.5, 3.6, 3.7, 3.8 show that VPBRs and VPFRs are suitable to construct the weakly nested reductions. Meanwhile, Proposition 3.9 shows that VQRS is not suitable to construct the nested reductions or the weakly nested reductions. All these results are very helpful for the algorithm designed to find a proper reduct quickly.

4. Some quick reduction algorithms

In this section, by using the nested structure of reduction, we design some algorithms to find a different reduct quickly when a reduct is given. To fasten the proposed algorithms, we improve them by finding the different reducts with no repetition. The existing algorithms to find the suboptimal reduct are also given.

4.1. Algorithms to find a different reduct based on the nested reduction

In this subsection, based on the nested reduction, we find a new reduct based on the given reduct or the obtained reduct, which is the main difference between the proposed algorithms in this paper and the existing algorithms. Because there are two types of nested structures (the forward nested reduction and the reverse nested reduction), it is necessary to propose two types of algorithms to find the nested reduction. As a result, based on the structure of the nested reduction, we design six algorithms (see Algorithms 1–6) called Nested Reduction Approaches. **DF** and **DR** are the nested reduction approaches designed by the dependency function, and **MF** and **MR** are the nested reduction approaches designed by the discernibility matrix. **FF** and **FR** are the nested reduction approaches designed by the flexible criteria of reduction.

In the following algorithms, $S = (U, A, D)$ represents the decision system. U represents the Universe, A represents all the condition attributes, D represents the decision attributes, and $RED(\beta)$ represents the given reduction on the threshold β .

Algorithm 1. Dependency function, Forward nested, DF for short).

Input: $S = (U, RED(\beta), D)$, the given threshold β and a new threshold α .

Output: $RED(\alpha)$.

Initialize: $RED(\alpha) = \phi$, $Candidate = RED(\beta)$

Step 1: Compute the dependency degree of $RED(\beta)$: $Dep(RED(\beta))$;

Step 2: Do

(2.1) For every $a_i \in Candidate$, compute the dependency degree of $RED(\alpha) \cup a_i$: $Dep(RED(\alpha) \cup a_i)$;

(2.2) Find a^* such that $Dep(RED(\alpha) \cup a^*)$ is maximal;

(2.3) If $Dep(RED(\alpha) \cup a^*) \leq Dep(RED(\beta))$, then let $RED(\alpha) = RED(\alpha) \cup a^*$, $Candidate = RED(\beta) - RED(\alpha)$;

While ($Dep(RED(\alpha) \cup a^*) < Dep(RED(\beta))$)

Step 3: Output $RED(\alpha)$

Algorithm 2 (Dependency function and Reverse nested, DR for short).

Input: $S = (U, A, D)$, the reduction $RED(\beta)$, the given threshold β and a new threshold α .
Output: $RED(\alpha)$
Initialize: $RED(\alpha) = RED(\beta)$
 Step 1: Compute the dependency degree of $RED(\alpha)$: $Dep(RED(\alpha))$, the dependency degree of A : $Dep(A)$;
 Step 2: if $Dep(RED(\alpha)) = Dep(A)$, goto Step 4, otherwise, goto Step 3;
 Step 3: Do
 (3.1) For every $a_i \in A - RED(\alpha)$, compute the dependency degree of $RED(\alpha) \cup a_i$: $Dep(RED(\alpha) \cup a_i)$;
 (3.2) **Find** a^* such that $Dep(RED(\alpha) \cup a^*)$ is maximal;
 (3.3) If $Dep(RED(\alpha) \cup a^*) \leq Dep(A)$, then let $RED(\alpha) = RED(\alpha) \cup a^*$;
 While $(Dep(RED(\alpha) \cup a^*) < Dep(A))$
 Step 4: If $RED(\alpha)$ is not independent, delete the redundant elements in $RED(\alpha)$; *Output* $RED(\alpha)$

Algorithm 3. Discernibility matrix, Forward nested, MF for short).

Input: $S = (U, RED(\beta), D)$, the given threshold β and a new threshold α .
Output: $RED(\alpha)$
Initialize: $RED(\alpha) = \phi$
 Step 1: Compute the lower approximation $R_{\beta, \alpha}([x]_D)$ for every $x \in U$ (here $R = RED(\beta)$).
 Step 2: Compute $c_{ij} = \{a : T(a(x_i, x_j), \lambda) \leq \alpha\}$, $\lambda = R_{\beta, \alpha}[x_i]_D(x_i)$ for $D(x_i, x_j) = 0$; Otherwise $c_{ij} = \phi$;
 Step 3: For every $c_{ij} \neq \phi$, if $c_{ij} \cap RED(\beta) \neq \phi$, let $RED(\alpha) = RED(\beta)$ and goto Step 5, otherwise goto Step 4.
 Step 4: While (there exist some $c_{ij} \neq \phi$)
 (4.1) Find the attribute a with the maximum frequency of occurrence in all $c_{ij} \neq \phi$, and add it into $RED(\alpha)$;
 (4.2) Delete those c_{ij} with nonempty overlap with a ;
 Step 5: If $RED(\alpha)$ is not independent, delete the redundant elements in $RED(\alpha)$; *Output* $RED(\alpha)$.

Algorithm 4. Discernibility matrix and Reverse nested, MR for short).

Input: $S = (U, A, D)$, the reduction $RED(\beta)$, the given threshold β and a new threshold α .
Output: $RED(\alpha)$
Initialize: $RED(\alpha) = RED(\beta)$
 Step 1: Compute the lower approximation $A_{\beta, \alpha}([x]_D)$ for every $x \in U$;
 Step 2: Compute $c_{ij} = \{a : T(a(x_i, x_j), \lambda) \leq \alpha\}$, $\lambda = A_{\beta, \alpha}[x_i]_D(x_i)$ for $D(x_i, x_j) = 0$; Otherwise $c_{ij} = \phi$;
 Step 3: Delete those c_{ij} with nonempty overlap with $RED(\alpha)$;
 Step 4: While (there exist some $c_{ij} \neq \phi$)
 (4.1) Find the attribute a with the maximum frequency of occurrence in all $c_{ij} \neq \phi$, and add it into $RED(\alpha)$;
 (4.2) Delete those c_{ij} with nonempty overlap with a ;
 Step 5: If Reduct is not independent, delete the redundant elements in $RED(\alpha)$; *Output* $RED(\alpha)$.

Algorithm 5. Flexible criteria measure, Forward nested, FF for short).

Input: $S = (U, A, D)$, the reduct $RED(\beta)$, the given threshold β and a new threshold α .
Output: $RED(\alpha)$.
Initialize: $RED(\alpha) = []$
 Step 1: Compute the dependency degree of A : $Dep(A)$;
 Step 2: Do
 (2.1) For every $a_i \in RED(\beta) - RED(\alpha)$, compute the dependency degree of $RED(\alpha) \cup a_i$: $Dep(RED(\alpha) \cup a_i)$;
 (2.2) Find a^* such that $Dep(RED(\alpha) \cup a_i)$ is maximal; let $RED(\alpha) = RED(\alpha) \cup a^*$;
 While $(Dep(A) - Dep(RED(\alpha)) > \alpha)$
 Step 3: *Output* $RED(\alpha)$.

Algorithm 6. Flexible criteria measure and Reverse nested, FR for short).

Input: $S = (U, A, D)$, the reduct $RED(\beta)$, the given threshold and β a new threshold α .
Output: $RED(\alpha)$
Initialize: $RED(\alpha) = RED(\beta)$
 Step 1: Compute the dependency degree of $RED(\alpha)$: $Dep(RED(\alpha))$, the dependency degree of A: $Dep(A)$;
 Step 2: If $Dep(A) - Dep(RED(\alpha)) \leq \alpha$, goto Step 4, otherwise goto Step 3.
 Step 3: Do
 (3.1) For every $a_i \in A - RED(\alpha)$, compute the dependency degree of $RED(\alpha) \cup a_i$: $DEP(RED(\alpha) \cup a_i)$;
 (3.2) **Find** a^* such that $Dep(RED(\alpha) \cup a_i)$ is maximal; let $RED(\alpha) = RED(\alpha) \cup a^*$;
 While $(Dep(A) - Dep(RED(\alpha))) > \alpha$
 Step 4: If $RED(\alpha)$ is not independent, delete the redundant elements in $RED(\alpha)$; *Output* $RED(\alpha)$

These algorithms are suited not only for the nested reduction but also for the weakly nested reduction. The main difference lies in the step of ‘checking independence’. The weakly nested reduction requires more time to check the ‘independence’ than does the nested reduction.

In the aforementioned algorithms, **DF**, **MF** and **FF** share the common idea, and all of them are designed based on the forward nested structure. The common designed idea is presented as follows:

Step 1: Given the threshold beta and alpha, beta here is smaller than alpha.
 Step 2: Take the reduct on beta as the candidate attribute set: CA.
 Step 3: Compute the dependency degree/discernibility matrix/criterion on CA.
 Step 4: Find the reduct on alpha from CA based on the increased searching strategy.

In Step 4, the theoretical basis is the forward nested structure. For this reason, these approaches are called forward nested algorithms.

DR, **MR** and **FR** share another common idea, and all of them are designed based on the reverse nested structure. The common designed idea is presented as follows:

Step 1: Given the threshold beta and alpha, beta here is larger than alpha.
 Step 2: Initialize the reduct on alpha being equal to the reduct on beta.
 Step 3: Compute the dependency degree/discernibility matrix/criterion on the whole universe U.
 Step 4: Find the reduct on alpha from U using an increased searching strategy.

In Step 4, the theoretical basis is the reverse nested structure. For this reason, these approaches are called reverse nested algorithms.

Based on the above explanations, the obvious difference between these two types of approaches is that the different theoretical bases are used: forward nested structure and reverse nested structure. Another obvious difference is that the discussed universes are different. The former approach takes the given reduct as the discussed universe; the latter takes the whole universe as the discussed universe.

These algorithms show that it is fast to find a new reduct when two thresholds, the reduct on one of which is known, are given. However, it is difficult to know whether the reducts are the same when two different thresholds are given. As a result, it is necessary to propose some algorithms that can display to what extent the reducts are the same.

4.2. Two improved algorithms to find different reducts based on nested reduction

In this subsection, we improve the above proposed algorithms to determine the critical point value on which the reduct changes. By using these algorithms, the reducts on all possible thresholds are obtained. The algorithm MALL is designed based on the perfect nested structure existing in discernibility matrix. The algorithm FALL is designed based on the perfect nested structure of flexible-criteria-based approaches. The dependence-function-based approach is not suitable to design the improved algorithms because only the weakly nested reduction exists in them.

Algorithm 7. Discernibility matrix, MALL for short).

Input: $S = (U, A, D)$, $A = \{a_1, \dots, a_k, \dots, a_r\}$.
Output: All_RED.
 Step 1: Compute the similarity degree of A and the lower approximation $A_{\vartheta}([x]_D)$ for every $x \in U$;
 Step 2: For $k = 1, \dots, r$, compute $fc_{ij}(k) = \min\{\beta : \vartheta(a_k(x_i, x_j), \beta) \leq A_{\vartheta_p}([x]_D)(x_i)\}$ for $D(x_i, x_j) = 0$; others $fc_{ij}(k) = 0$;

Step 3: For any i, j, k , sort $fc_{ij}(k)$ in ascending order and save in a vector named as *Threshold*; Let $t = 0$, $RED(t) = \phi$;

Step 4: Do while ($t < \text{length}(\text{Threshold})$)

(4.1) Let $t = t + 1$, $\gamma = \text{Threshold}(t)$, and compute γ -cut of each fc_{ij} , denoted as $(fc_{ij})_\gamma$;

(4.2) Let $RED(t) = RED(t - 1)$, Delete those $(fc_{ij})_\gamma$ with nonempty overlap with $RED(t)$;

(4.3) While (there exist some $(fc_{ij})_\gamma \neq \phi$)

(4.3.1) Add the attribute a_k with maximum frequency of occurrence in all $(fc_{ij})_\gamma \neq \phi$ into $RED(t)$;

(4.3.2) Delete those $(fc_{ij})_\gamma$ with nonempty overlap with a_k ;

(4.4) If $RED(\gamma)$ is not independent, delete the redundant elements in $RED(t)$;

(4.5) Let $All_RED(t) = (RED(t), \gamma)$;

(4.6)

(4.6.1) compute the intersection of the matrix on $RED(t)$

(4.6.2) find the maximum 'alpha' in the intersected matrix

(4.6.1) find the position 't' of 'alpha' staying in 'Threshold'

Step 5: Output *All_RED*.

In MALL, it is critical to find all of the boundary values on which the reduct size changes. There are three steps to determine these very boundary values: compute the fuzzy discernibility matrix, find all possible boundary values, and find the very boundary values. In the following section, we will describe these steps clearly.

4.2.1. Computing the robust fuzzy discernibility matrix

Suppose $(fc_{ij})_{n \times n}$ is the fuzzy discernibility matrix. Each entry in the fuzzy discernibility matrix is defined as $fc_{ij}(k) = \min\{\beta : \vartheta(a_k(x_i, x_j), \beta) \leq A_{\vartheta_\beta}([x]_D)(x_i)\}$, $\forall k \in R$, where $\beta \in [0, 1]$. In the real applications, the fuzzy discernibility matrix is obtained based on the specific triangular norm T . Let $\vartheta = \vartheta_{T, \cdot}$, for example, then

$$fc_{ij}(k) = \begin{cases} 0, & D(x_i, x_j) = 1, \\ a_k(x_i, x_j), & a_k(x_i, x_j) > \max_{y \in U} (R(x_i, y) \text{ and } D(x_i, x_j) = 0), \\ 0, & a_k(x_i, x_j) \leq \max_{y \in U} (R(x_i, y) \text{ and } D(x_i, x_j) = 0). \end{cases}$$

4.2.2. Finding all possible boundary values

All possible boundary values can be found from the robust fuzzy discernibility matrix. All membership degrees of the fuzzy sets in the robust fuzzy discernibility matrix are composed of the set of all possible boundary values. We obtain all these values by using the following steps:

- (1) Collect all the membership degrees of the fuzzy sets in the fuzzy discernibility matrix.
- (2) Save those values in ALLPV.
- (3) Find the values in ALLPV with no repetitions and save them in PV.
- (4) Sort the values increasingly in PV.

4.2.3. Finding the quite boundary values and the corresponding reducts

It is critical to find the boundary values from PV on which the reduct size changes. We design the following steps to find all boundary values.

- (1) Set the original value: $\alpha = PV(1)$ and $s = 1$.
- (2) While $\alpha \leq \max(PV)$:
 - (2.1) Let $VPV(s) = \alpha$ and find the α -reduct ' RED_α '.
 - (2.2) For $k = 1:\text{length}(RED_\alpha)$:
 - (2.2.1) Find the left attributes $K = RED_\alpha - a_k$ in RED_α .
 - (2.2.2) Compute $\max_{a_k \in K}(\min_{ij} fc_{ij}(s))$, and save it as $Thr(k)$.
 - (2.3) Compute $\min(Thr)$ and save it as α , Let $s = s + 1$.
- (3) VPV is the set of all the boundary values on which the reduct size changes.

Clearly, it is easy to see that the reducts on all of the possible parameters can be found when all the boundary values are selected.

Algorithm 8. Flexible criteria measure to find reducts on all levels of approximation, FALL for short).

Input: $S = (U, A, D)$.

Output: All_RED .

Initialize: $Candidate = \phi$

Step 1: Compute the similarity degree of A , the dependency degree of A : $Dep(A)$; Let $k = 0$;

Step 2: Do

(2.1) For every $a_i \in A$, compute the dependency degree of $Candidate \cup a_i$: $Dep(Candidate \cup a_i)$;

(2.2) **Find** a^* such that $Dep(Candidate \cup a^*)$ is maximal;

(2.4) Let $\beta = Dep(A) - Dep(Candidate \cup a^*)$, $A = A - a^*$ and $Candidate = Candidate \cup a^*$;

(2.5) If $Candidate$ is not independent with respect to β , delete the redundant elements from $Candidate$;

(2.6) Let $k = k + 1, All_RED(k) = (Candidate, \beta)$.

While $(Dep(A) - Dep(Candidate \cup a^*) \neq 0)$.

Step 3: Output All_RED .

In the dependence-function-based models, the boundary values are easy to determine because the dependency function is monotonic with the increment of attributes. As a result, this idea is used in Step 4 of FALL.

MALL and FALL are the first to be proposed to find the reducts on all possible thresholds. Now all the existing algorithms are proposed on one fixed threshold. Then, the intuitive way to find the reducts on all possible thresholds is to execute the existing algorithms on all possible thresholds. Nevertheless, it is clear that this intuitive method is NP-complete. Comparatively, MALL and FALL are not NP-complete because the designation of MALL and FALL can find the critical point values on which the reducts change. We list more detailed reasons as follows.

- (1) The first and most important reason is that MALL and FALL can determine the critical point value on which the reduct changes. The nested structure shows that, to some extent of the thresholds, the reduct does not change. Based on this discovery, we design methods to determine the critical point values, and then the NP problem becomes a polynomial problem.
- (2) MALL and FALL avoid some repetition in computing reducts on different thresholds. Assume that all of the critical point values are given. An intuitive algorithm is used to execute the existing algorithms on all the critical point values. This intuitive algorithm is not NP-complete, but it is much slower than MALL and FALL. Because MALL and FALL use the nested structure to avoid repetition in computing the reducts.

4.3. The existing heuristic to find a reduct

In this subsection, we present the existing approaches ME, DE and FE (see Algorithms 9–11). Formally, these algorithms seem similar to the algorithms of the nested reduction approaches. The main difference between the existing algorithms and the nested reduction approaches is whether the reduct obtained is used in the algorithms.

Algorithm 9. Dependency function, DE for short).

Input: $S = (U, A, D)$ and the threshold α .

Output: $RED(\alpha)$.

Initialize: $RED(\alpha) = \phi$

Step 1: Compute the dependency degree of A : $Dep(A)$; Let $RED(\alpha) = \phi$

Step 2: Do

(2.1) For every $a_j \in A$, compute its dependency degree of $RED(\alpha) \cup a_j$: $Dep(RED(\alpha) \cup a_j)$;

(2.2) **Find** a^* such that $Dep(RED(\alpha) \cup a^*)$ is maximal;

(2.3) **Let** $A = A - a^*$ and $RED(\alpha) = RED(\alpha) \cup a^*$;

While $(Dep(RED(\alpha) \cup a^*) < Dep(A))$

Step 3: If Reduct is not independent, delete the redundant elements in $RED(\alpha)$; then Output $RED(\alpha)$.

Algorithm 10. Discernibility matrix, ME for short).

Input: $S = (U, A, D)$ and the threshold α .

Output: $RED(\alpha)$

Initialize: $RED(\alpha) = \phi$

- Step 1: Compute the similarity degree of A and the lower approximation $\underline{A}_{\alpha}([x]_D)$ for every $x \in U$;
- Step 2: Compute $c_{ij} = \{a : T(a(x_i, x_j), \lambda) \leq \alpha\}$, $\lambda = \underline{A}_{\alpha}([x_i]_D)(x_i)$ for $D(x_i, x_j) = 0$; Otherwise $c_{ij} = \phi$;
- Step 3: Compute $Core_{D_{\alpha}}(A)$; Delete those c_{ij} with nonempty overlap with $Core_{D_{\alpha}}(A)$; Let $RED(\alpha) = Core_{D_{\alpha}}(A)$;
- Step 4: While (there exist some $c_{ij} \neq \phi$)
 - (4.1) Add the attribute a with maximum frequency of occurrence in all $c_{ij} \neq \phi$ into $RED(\alpha)$;
 - (4.2) Delete those c_{ij} with nonempty overlap with a ;
- Step 5: If Reduct is not independent, delete the redundant elements in $RED(\alpha)$; then output $RED(\alpha)$.

Algorithm 11. Flexible criteria measure, FE for short).

- Input:* $S = (U, A, D)$, the threshold α .
Output: $RED(\alpha)$
Initialize: $RED(\alpha) = \phi$
- Step 1: Compute the dependency degree of A: $Dep(A)$;
 - Step 2: Do
 - (2.1) For every $a_i \in A - RED(\alpha)$, compute the dependency degree of $RED(\alpha) \cup a_i$; $Dep(RED(\alpha) \cup a_i)$;
 - (2.2) **Find** a^* such that $Dep(RED(\alpha) \cup a^*)$ is maximal; let $RED(\alpha) = RED(\alpha) \cup a^*$;
 While ($Dep(A) - Dep(RED(\alpha)) > \alpha$)
 - Step 3: If $RED(\alpha)$ is not independent, delete the redundant elements in $RED(\alpha)$; Output $RED(\alpha)$

4.4. Scalability analysis

In Tables 1 and 2, we give the detailed time complexity comparison. We find that, although the nested reduction approaches do not change the level of the time complexity, they significantly reduce the size of the time complexity.

When the reduct that has been obtained is used in the algorithms, the input dataset or the initialized reduct is changed. In Algorithms 1, 3 and 5, the size of the input dataset is significantly reduced to the size of the given reduct. In Algorithms 2, 4 and 6, the initialized reduct is changed from the empty set to the given reduct. As a result, the nested reduction approaches are far quicker than the approaches that do not use the nested reduction. We will demonstrate this fact in Section 5.

In Table 2, k is the execution time of the algorithms FE and ME, which needs to be set beforehand. The parameter t is the number of all boundary values. In general, k is larger than t , especially if the step length is small. The parameter $|P|$ is the number of the reducts obtained in the previous boundary values. As a result, $|P|$ becomes smaller with the increment of t . The time complexity of FALL and MALL is obviously smaller than FE and ME.

5. Numerical experiments

In this section, several datasets with different object numbers, decision classes, condition attributes and data distributions (see Table 3) are selected to demonstrate the performance of the algorithms designed based on the nested reduction.

As mentioned previously in Section 3, there are many parameterized models suited for the construction of the nested or weakly nested reductions. Here, FVPRS, VPBRs and fuzzy-RED, which determine reducts by using three different types of attribute reduction approaches, are selected as examples to perform some demonstrations. Similar results can be obtained on the other parameterized models suited for the construction of the nested or weakly nested reductions.

Table 1
the detailed information of the time complexity in Algorithms 1, 2, 3, 4, 5, 6, 9, 10, 11.

	Designed by discernibility matrix			Designed by dependency function and designed by flexible criteria of reduction			
	Using the nested reduction or not	No	Yes	Using the nested reduction or not	No	Yes	
	<i>ME</i>	<i>MF</i>	<i>MR</i>		<i>DE,FE</i>	<i>DF,FF</i>	<i>DR,FR</i>
Similarity Matrix	$O(U ^2 \times A)$ $O(U ^2 \times A)$	$O(U ^2 \times RED(\beta))$ $O(U ^2 \times RED(\beta))$	$O(U ^2 \times A)$ $O(U ^2 \times A)$	Similarity Dependency (or quality measure)	$O(U ^2 \times A)$ $O(U)$	$O(U ^2 \times RED(\beta))$ $O(U)$	$O(U ^2 \times A)$ $O(U)$
Loop All	$O(U ^2 \times A ^2)$ $O(U ^2 \times A ^2)$	$O(U ^2 \times RED(\beta) ^2)$ $O(U ^2 \times RED(\beta) ^2)$	$O(U ^2 \times (A - RED(\beta))^2)$ $O(U ^2 \times (A - RED(\beta))^2)$	Loop All	$O(U ^2 \times A ^2)$ $O(U ^2 \times A ^2)$	$O(U ^2 \times RED(\beta) ^2)$ $O(U ^2 \times RED(\beta) ^2)$	$O(U ^2 \times (A - RED(\beta))^2)$ $O(U ^2 \times (A - RED(\beta))^2)$

Table 2
the detailed information of time complexity in Algorithms 7–11.

Designed by discernibility matrix			Designed by flexible criteria		
Using the nested reduction or not	No ME	Yes MALL	Using the nested reduction or not	No FE	Yes FALL
Similarity	$O(k \times U ^2 \times A)$	$O(U ^2 \times A)$	Similarity	$O(k \times U ^2 \times A)$	$O(U ^2 \times A)$
Matrix	$O(k \times U ^2 \times A)$	$O(t \times U ^2 \times P)$	Dependency (or quality measure)	$O(k \times U)$	$O(U)$
Loop	$O(k \times U ^2 \times A ^2)$	$O(t \times U ^2 \times \frac{ P (P -1)}{2})$	Loop	$O(k \times U ^2 \times A ^2)$	$O(U ^2 \times A ^2)$
All	$O(k \times U ^2 \times A ^2)$	$O(t \times U ^2 \times \frac{ P (P -1)}{2})$	All	$O(k \times U ^2 \times A ^2)$	$O(U ^2 \times A ^2)$

Table 3
The information of some datasets from UCI [35].

Datasets	Abbr.	Objects	Condition	Data type	Classes
Iris	Iris	150	4	Real value	3
Diabetes	Diab	769	8	Real value	2
Ionosphere	Ionos	351	34	Real value	2
Wine	Wine	178	13	Real value	3
Sonar	Sonar	208	60	Real value	2
Wdbc	Wdbc	569	30	Real value	2
Wdbc	Wdbc	198	32	Real value	2
New_thyroid	New_thy	215	5	Real value	3
Haberman	Haberman	306	3	Real value	2

In the following, we first demonstrate the structure of the nested reduction by numerical results. Then, we by experimental results show that the nested reduction approaches perform obviously or even significantly better than those algorithms without using the nested reduction.

5.1. The structure of the nested reduction

Here we take FVPRS as an example to demonstrate the structure of the nested reduction. And the improved algorithm to find reducts, i.e., MALL, is selected to perform the demonstration. The numerical results are presented in Tables 4–6. These tables clearly demonstrate that the reducts act as a nested structure with the incremental parameter.

Table 4
The nested reduction in FVPRS (diabetes, iris and new_thyroid).

Diabetes									Iris					New_thyroid						
Parameters	Reduct								Parameters	Reduct				Parameters	Reduct					
0	1	2	3	4	5	6	7	8	0	1	2	3	4	0	1	2	3	4	5	
0.95	1	2	3	4	6	7	8	0.945	1	2	4	0.911	1	2	3	5				
0.951	1	2	3	4	7	8					0.972	1	2	5						
0.983	1	2	4	7	8															
0.99	2	4	7	8																
0.997	2	7	8																	

Table 5
The nested reduction in FVPRS (wine).

Wine													
Parameter	Reduct												
0	1	2	3	4	5	6	7	8	9	10	11	12	13
0.765	1	2	3	5	6	7	8	9	10	11	12	13	
0.791	1	2	3	6	7	8	9	10	11	12	13		
0.793	1	2	3	6	7	8	9	10	12	13			
0.821	1	2	3	6	8	9	10	12	13				
0.828	1	2	3	6	8	9	12	13					
0.857	1	2	3	6	9	12	13						
0.878	1	2	3	9	12	13							
0.899	1	3	9	12	13								
0.949	3	9	12	13									
0.978	3	9	13										
0.995	3	9											

Table 6
The nested reduction in FVPRS (wdbc).

WPBC parameters	Reduct
0	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32
0.664	1 2 3 4 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32
0.707	1 2 3 4 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 26 27 28 29 30 31 32
0.746	1 2 3 4 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 26 27 28 29 30 31 32
0.761	1 2 3 4 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 26 28 29 30 31 32
0.776	1 2 3 4 6 7 8 9 10 11 12 13 14 16 17 18 19 20 21 22 23 26 28 29 30 31 32
0.781	1 2 3 4 6 7 8 9 10 11 12 13 14 16 17 18 19 20 21 22 23 26 28 29 30 32
0.784	1 2 3 4 6 7 8 9 10 12 13 14 16 17 18 19 20 21 22 23 26 28 29 30 32
0.786	1 2 3 4 6 7 8 9 10 12 13 14 16 17 18 19 20 22 23 26 28 29 30 32
0.792	1 2 3 4 6 7 8 9 10 12 13 14 17 18 19 20 22 23 26 28 29 30 32
0.798	1 2 3 4 6 8 9 10 12 13 14 17 18 19 20 22 23 26 28 29 30 32
0.823	1 2 4 6 8 9 10 12 13 14 17 18 19 20 22 23 26 28 29 30 32
0.83	1 4 6 8 9 10 12 13 14 17 18 19 20 22 23 26 29 30 32
0.831	1 4 6 8 10 12 13 14 17 18 19 20 22 23 26 29 30 32
0.834	1 4 6 8 10 12 13 17 18 19 20 22 23 26 29 30 32
0.835	1 4 6 8 10 12 13 17 19 20 22 23 26 29 30 32
0.844	1 4 6 8 10 12 13 17 20 22 23 26 29 30 32
0.847	1 4 8 10 12 13 17 20 22 23 26 29 30 32
0.858	1 8 10 12 13 20 22 23 26 29 30 32
0.863	1 8 10 12 13 20 22 23 26 29 32
0.865	1 10 12 13 20 22 23 26 29 32
0.871	1 10 12 13 22 23 26 29 32
0.881	1 10 13 22 23 26 29 32
0.896	1 10 13 22 23 26 32
0.919	1 10 13 22 26 32
0.944	1 10 13 26 32
0.968	1 10 13 26
0.984	1 10 26
0.995	1 10

5.2. To compare the algorithms of using nested reduction or not: finding one reduct

This subsection focuses on the comparison of finding a different reduct when a reduct is given. The comparison is setup as follows:

- (1) The parameters are chosen on the interval [0,1] with the step length 0.01.
- (2) The running times, as index, are compared.
- (3) Three types of the nested reduction algorithms and the existing algorithms are compared.
- (4) FVPRS, VPBRS and fuzzy-RED are selected as examples.

The comparison results are summarized in Figs. 1–3.

In the above figures, MR and MF are the algorithms that use the nested reduction. M is the existing algorithm, which does not use the nested reduction. These figures show that the algorithms using the nested reduction are often obviously faster than the algorithms that do not use the nested reduction. It is thus easy to draw a conclusion that the nested reduction is very helpful in finding a different reduct quickly when a reduct is given.

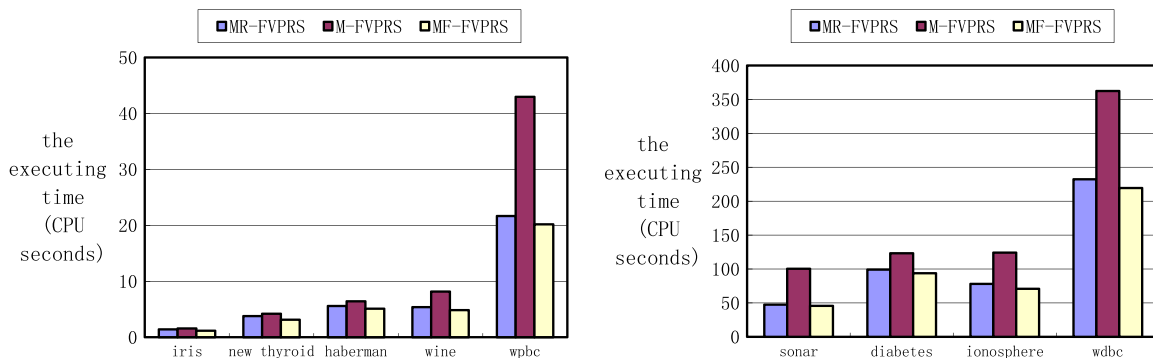


Fig. 1. The comparison of running time: FVPRS.

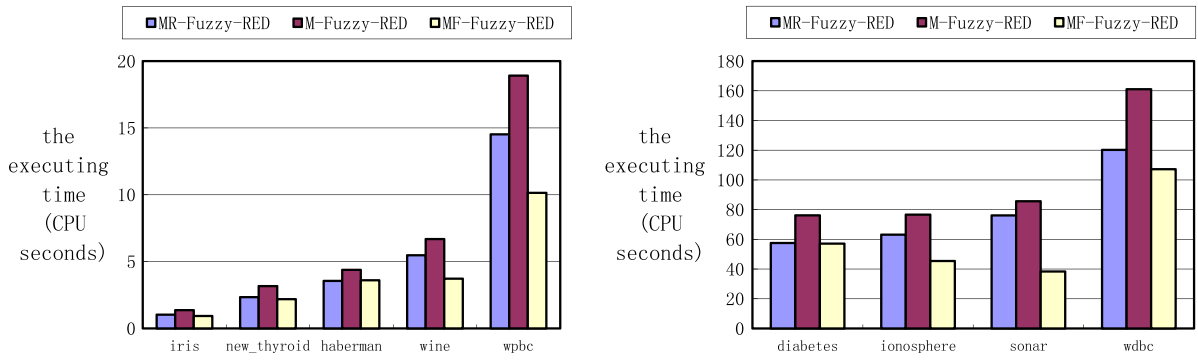


Fig. 2. The comparison of running time: Fuzzy-RED.

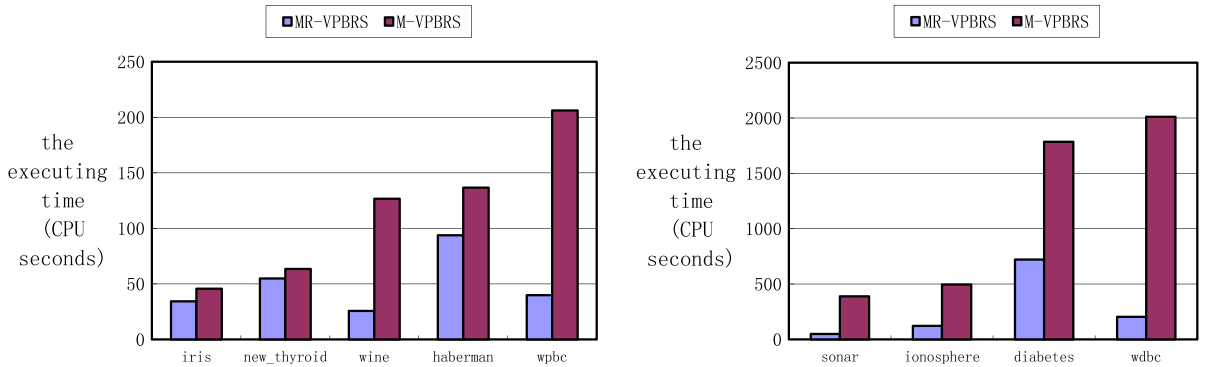


Fig. 3. The comparison of running time: VPBR.

5.3. To compare the improved algorithms with the existing algorithms

In this subsection, we compare the improved algorithms, i.e., MALL and FALL, and the existing algorithms, i.e., ME and FE. First, we highlight the limitations and advantages of these algorithms. These existing approaches have one limitation: they often find many repeated reducts. As a result, these existing approaches are time-consuming, and many similar computations are repeated many times using the existing approaches. The improved algorithms that use the nested reduction have one obvious advantage: by using the improved algorithms, those repeated computations can be skipped, and then the computing time is saved.

In the following, we experimentally show differences between these algorithms. The experiments are set up as follows:

- (1) The algorithms, ME and FE, are executed on the interval [0,1] with the step length 0.01.
- (2) The indices, including the running time, the repeated reducts and the obtained reducts, are compared.
- (3) FVPRS and Fuzzy-RED are selected as examples to compare with the existing algorithms.

Tables 7 and 8 display one obvious fact: MALL and FALL do not obtain repeated reducts, but ME and FE do. Figs. 4 and 5 clearly show that MALL and FALL are significantly faster than the existing algorithm. All these facts show that as the heuristic

Table 7
Comparison of obtained reducts in FVPRS.

	The number of all reducts		The number of repeated reducts		The number of obtained reducts ^a	
	MALL	ME	MALL	ME	MALL	ME
Diabetes	6	99	0	95	6	4
Haberman	1	99	0	98	1	1
Ionos	29	99	0	76	29	23
Iris	2	99	0	97	2	2
New_thyroid	3	99	0	96	3	3
Sonar	50	99	0	67	50	32
WDBC	29	99	0	81	29	18
Wine	12	99	0	90	12	9
WPBC	29	99	0	78	29	21

^a Here the number of the obtained reduct means the number of reducts when the repeated reducts are deleted.

Table 8
Comparison of obtained reducts in Fuzzy-RED.

	The number of all reducts		The number of repeated reducts		The number of obtained reducts ^a	
	FALL	FE	FALL	FE	FALL	FE
Diabetes	8	99	0	91	8	8
Haberman	3	99	0	93	3	3
Ionos	33	99	0	84	33	15
Iris	4	99	0	95	4	4
New_thyroid	5	99	0	94	5	5
Sonar	57	99	0	76	57	23
WDBC	30	99	0	88	30	11
Wine	13	99	0	88	13	11
WPBC	32	99	0	85	32	14

^a Here the number of the obtained reduct means the number of reducts when the repeated reducts are deleted.

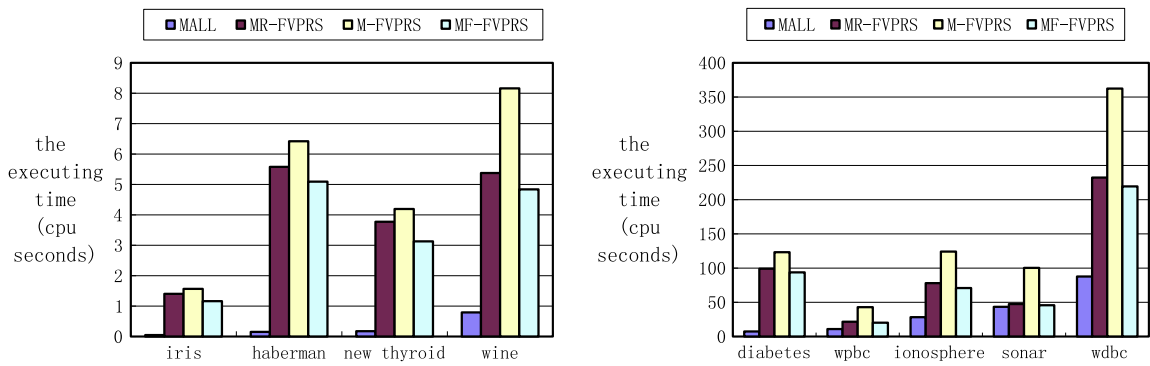


Fig. 4. The Comparison of running times in FVPRS.

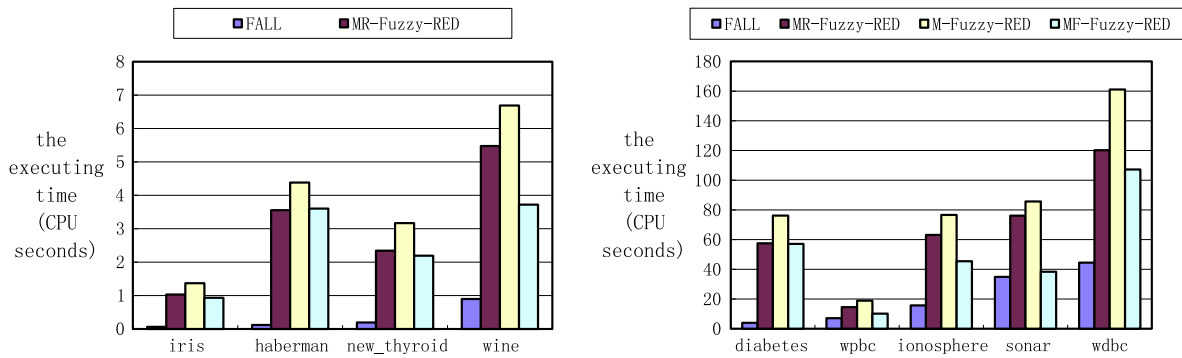


Fig. 5. The comparison of running times in Fuzzy-RED.

algorithms, MALL and FALL, often find more reducts without repetition. Comparatively, the lazy algorithms, ME and FE, often do a significant amount of repeated work and are time-consuming.

6. Conclusions

This paper studies the structure of the reducts on the different parameters in parameterized rough sets. The authors discover that the reducts on different parameters are often nested or the same. In this paper, it is called the nested structure. Based on this interesting discovery, some theorems and algorithms have been proposed. The theorems show, by using strict mathematical reasoning, that the nested structure exactly exists in parameterized rough reduction. Using several theorems and several properties, we present a systematic framework that clearly describes the nested structure of reducts in parameterized rough reduction. The algorithms show that the nested structure is very useful and effective in real applications. By

using the nested structure, some feasible and effective algorithms have been designed that find a proper reduct quickly. In particular, MALL and FALL are the first to be proposed to find the reducts on all of the possible thresholds.

Acknowledgments

This research is supported by the National Basic Research Program of China (973 Program) (2012CB316205), the National Science Foundation of China (61202114, 61272137, 61070056, 61033010, 61170040 and 60903089). This research is also supported by the Fundamental Research Funds for the Central Universities, and the Research Funds of Renmin University of China (12XNLF07). We also acknowledge the anonymous reviewers for their valuable comments and suggestions.

Appendix A

In this part, we will briefly recall the notions of rough sets, fuzzy sets and fuzzy logical operators. Interested readers are referred to [7,8,22,44] for further details and backgrounds.

A.1. Rough sets

The theory of rough sets proposed by Pawlak is an extension of set theory for the study of intelligent systems characterized by insufficient and incomplete information. It is formally based on an information system which is a pair $A = (U, A)$, where U is a non-empty finite set of objects called the Universe and A is a non-empty finite set of attributes. With every subset of attributes $B \subseteq A$, we associate a binary relation defined as $IND(B) = \{(x, y) \in U : a(x) = a(y), \forall a \in B\}$. The sets $\{x \in U : [x]_B \subseteq X\}$ and $\{x \in U : [x]_B \cap X \neq \emptyset\}$ are called B -lower and B -upper approximation of X , denoted by $\underline{B}X$ and $\overline{B}X$, respectively. By using these approximation operators, knowledge hidden in data can be represented as possible and certain rules.

A.2. Fuzzy logic operators

Fuzzy logic operators play an important role in the development of fuzzy rough set theory. We therefore recall some important definitions before briefly reviewing FRS. These definitions are: triangular norm (i.e., T -norm) and T -residuated implication. Other fuzzy logical operators are referred [22] for details.

A triangular norm, or shortly T -norm, is a function $T: [0, 1] \times [0, 1] \rightarrow [0, 1]$ that satisfies the following conditions: monotonicity (if $x < \alpha$, $y < \beta$, then $T(x, y) \leq T(\alpha, \beta)$), commutativity ($T(x, y) = T(y, x)$), associativity ($T(T(x, y), z) = T(x, T(y, z))$), and boundary condition ($T(x, 1) = x$). The most popular continuous T -norms include the standard T -norm: $T_M(x, y) = \min\{x, y\}$ and the Lukasiewicz T -norm: $T_L(x, y) = \max\{0, x + y - 1\}$.

Given a lower semi-continuous triangular norm T , the residuation implication, or called the T -residuated implication, is a function $\vartheta: [0, 1] \times [0, 1] \rightarrow [0, 1]$ that satisfies $\vartheta(x, y) = \sup \{z \in [0, 1], T(x, z) \leq y\}$ for every $x, y \in [0, 1]$. The T -residuated implications include the Godel implication ϑ_M which is based on T_M : $\vartheta_M = \begin{cases} 1, & x \leq y \\ y, & x > y \end{cases}$ and the Lukasiewicz implication ϑ_L which is based on T_L : $\vartheta_L = \min\{1 - x + y, 1\}$.

A.3. Fuzzy decision system

Let U , called the Universe, be a nonempty set with a finite number of objects. Each object is described by a set of fuzzy condition attributes, denoted by R , and decision attributes, denoted by D . The triple of $(U, R \cup D)$ is then called a fuzzy decision system, denoted by FD . Here, fuzzy attributes mean the attributes with real values that can be transferred into fuzzy values.

A.4. Fuzzy similarity relation

With every $P \subseteq R$, we associate a binary relation $P(x, y)$ called fuzzy similarity relation of P , which is a binary relation satisfying (s.t.) reflexivity ($P(x, x) = 1$), symmetry ($P(x, y) = P(y, x)$) and T -transitivity ($P(x, y) \geq T(P(x, z), P(z, y))$) for every $x, y, z \in U$. For simplicity, we use P to represent its similarity relation, $F(U)$ to represent the fuzzy power set of U .

A.5. Fuzzy rough sets

Fuzzy rough sets (FRS) are developed to handle databases with real values by combining fuzzy sets and rough sets. Fuzzy rough sets (FRS) were first proposed by Dubois and Prade [7,8] and then studied in detail in [22,27,44]. In general, the existing FRS can be summarized as the following four approximation operators: T - upper approximation operator: $\overline{R}_T A(x) = \sup_{u \in U} T(R(x, u), A(u))$; ϑ - lower approximation operator: $\underline{R}_\vartheta A(x) = \inf_{u \in U} \vartheta(R(x, u), A(u))$; s - lower approximation

operator: $\underline{R}_\sigma A(x) = \inf_{u \in U} \mathcal{N}(R(x, u), A(u))$; σ – upper approximation operator: $\overline{R}_\sigma A(x) = \sup_{u \in U} \sigma(N(R(x, u), A(u)))$. In this paper, we focus on the work based on T – upper and ϑ – lower approximation operators. Similar work can be obtained on the left two operators by using the duality.

References

- [1] S. An, Q.H. Hu, D.R. Yu, J.F. Liu, Soft minimum-enclosing-ball based robust fuzzy rough sets, *Fundam. Inform.* 115 (2–3) (2012) 189–202.
- [2] M. Beynon, An investigating of beta-reduct selection within the variable precision rough sets model, in: *The 2nd Int. Conf. on Rough Sets and Current Trend in Computing*, LNAI, 2000, pp. 114–122.
- [3] M. Beynon, Reducts within the variable precision rough sets model: a further investigation, *Eur. J. Oper. Res.* 134 (2001) 592–605.
- [4] C. Cornelis, M.D. Cock, A.M. Radzikowska, Vaguely quantified rough sets, *Rough Sets Fuzzy Sets Data Min. Granular Comput. – LNCS 4482* (2007) 87–94.
- [5] C. Cornelis, G.H. Martin, R. Jensen, D. Slezak, Feature Selection with Fuzzy Decision Reducts, *RSKT2008*, pp. 284–291.
- [6] C. Cornelis, R. Jensen, G.H. Martin, D. Slezak, Attribute selection with fuzzy decision reducts, *Inform. Sci.* 180 (2) (2010) 209–224.
- [7] D. Dubois, H. Prade, Rough fuzzy sets and fuzzy rough sets, *Int. J. Gen. Syst.* 17 (1990) 191–208.
- [8] D. Dubois, H. Prade, Putting rough sets and fuzzy sets together, in: R. Slowinski (Ed.), *Intelligent Decision support, Handbook of Applications and Advances of the Rough Sets Theory*, Kluwer Academic Publishers, 1992, pp. 203–232.
- [9] Q.H. Hu, S. An, D.R. Yu, Soft fuzzy rough sets for robust feature evaluation and selection, *Inf. Sci.* 180 (22) (2010) 4384–4400.
- [10] Q.H. Hu, L. Zhang, S. An, D. Zhang, D.R. Yu, On robust fuzzy rough set models, *IEEE Trans. Fuzzy Syst.* 20 (4) (2012) 636–651.
- [11] Q.H. Hu, Z.X. Xie, D.R. Yu, Hybrid attribute reduction based on a novel fuzzy-rough model and information granulation, *Pattern Recog.* 40 (2007) 3509–3521.
- [12] Q.H. Hu, D.R. Yu, Z.X. Xie, Information-preserving hybrid data reduction based on fuzzy-rough techniques, *Pattern Recog. Lett.* 27 (5) (2006) 414–423.
- [13] K.Y. Huang, Application of VPRS model with enhanced threshold parameter selection mechanism to automatic stock market forecasting and portfolio selection, *Expert Syst. Appl.* 36 (9) (2009) 11652–11661.
- [14] R. Jensen, Q. Shen, Semantics-preserving dimensionality reduction: rough and fuzzy-rough-based approaches, *IEEE Trans. Knowl. Data Eng.* 16 (12) (2004) 1457–1471.
- [15] R. Jensen, Q. Shen, Fuzzy-rough sets assisted attribute selection, *IEEE Trans. Fuzzy Syst.* 15 (1) (2007) 73–89.
- [16] J.D. Katzberg, W. Ziarko, Variable precision rough sets with asymmetric bounds, in: W. Ziarko (Ed.), *Rough Sets, Fuzzy Sets and Knowledge Discovery*, Springer, 1994, pp. 167–177.
- [17] X.Y. Jia, W.H. Liao, Z.M. Tang, L. Shang, Minimum cost attribute reduction in decision-theoretic rough set models, *Inf. Sci.* 219 (2013) 151–167.
- [18] D. Lzak, W. Ziarko, Attribute reduction in the bayesian version of variable precision rough set model, *Electron. Notes Theor. Comput. Sci.* 82 (4) (2003) 1–11.
- [19] A. Mieszkowicz-Rolka, L. Rolka, Variable precision fuzzy rough sets, *Trans. Rough Sets* 1 (2004) 144–160.
- [20] A. Mieszkowicz-Rolka, L. Rolka, Variable precision fuzzy rough sets model in the analysis of process data, in: *International Workshop on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing*, 2005, pp. 354–363.
- [21] M. Fan, W. Zhu, Attribute reduction of data with error ranges and test costs, *Inf. Sci.* 211 (2012) 48–67.
- [22] N.N. Morsi, M.M. Yakout, Axiomatics for fuzzy rough sets, *Fuzzy Sets Syst.* 100 (1998) 327–342.
- [23] H.S. Nguyen, D. Slezak, Approximate reducts and association rules-corresponding and complexity results, in: A. Skowron, S. Ohsuga, N. Zhong (Eds.), *Proceedings of the 7th International Workshop on Rough Sets, Fuzzy Sets, Data Mining, and Granular-Soft Computing*, 1999, pp. 137–145.
- [24] Z. Pawlak, Rough sets, *Int. J. Comput. Inf. Sci.* 11 (1982) 341–356.
- [25] Z. Pawlak, *Rough Sets: Theoretical Aspects of Reasoning about Data [M]*, Kluwer Academic Publishers, Dordrecht, 1991.
- [26] J.M.F. Salido, S. Murakami, rough set analysis of a general type of fuzzy data using transitive aggregation of fuzzy similarity relation, *Fuzzy Sets Syst.* 139 (2003) 635–660.
- [27] A.M. Radzikowska, E.E. Kerre, A comparative study of fuzzy rough sets, *Fuzzy Sets Sys.* 126 (2002) 137–155.
- [28] D. Slezak, Approximate reducts in decision tables, in: *IMPU-96 Information Processing and Management of Uncertainty on Knowledge-Based Systems [C]*, vol. 3, Granada, Spain, 1996, pp. 1159–1164 (1–5).
- [29] D. Slezak, Approximate entropy reducts, *Fundam. Inform.* 53 (2002) 365–390.
- [30] D. Slezak, Rough sets and functional dependencies in data: foundations of association reducts, *Trans. Comput. Sci.* 5 (2009) 182–205.
- [31] D. Slezak, W. Ziarko, Variable precision bayesian rough set model, in: *RSFDGrC 2003*, LNAI, vol. 2639, Springer Verlag, 2003, pp. 312–315.
- [32] D. Slezak, W. Ziarko, The investigation of the Bayesian rough set model, *Int. J. Approx. Reason.* 40 (1–2) (2005) 81–91.
- [33] C.T. Su, J.H. Hsu, Precision parameter in the variable precision rough sets model: an application, *OMEGA – Int. J. Manag. Sci.* 34 (2) (2006) 149–157.
- [34] C.C. Tsang, D.G. Chen, D.S. Yeung, X.Z. Wang, J.W.T. Lee, Attributes reduction using fuzzy rough sets, *IEEE Trans. Fuzzy Syst.* 16 (5) (2008) 1130–1141.
- [35] UCI, Machine Learning Repository, 2005. <<http://www.ics.uci.edu/~mllearn/MLRepository.html>>.
- [36] W.Z. Wu, Y. Leung, W.X. Zhang, Connections between rough set theory and Dempster-Shafer Theory of Evidence, *Int. J. Gen. Syst.* 31 (4) (2002) 405–430.
- [37] W.Z. Wu, M. Zhang, H.Z. Li, J.S. Mi, Knowledge reduction in random information systems via Dempster-Shafer theory of evidence, *Inf. Sci.* 174 (2005) 143–164.
- [38] Y.Y. Yao, Probabilistic rough set approximations, *Int. J. Approx. Reason.* 49 (2) (2008) 255–271.
- [39] Y.Y. Yao, Decision-theoretic rough set models, *LNCS 4481* (2007) 1–12.
- [40] Y.Y. Yao, Y. Zhao, Attribute reduction in decision-theoretic rough set models, *Inf. Sci.* 178 (17) (2008) 3356–3373.
- [41] Y.Y. Yao, Three-way decisions with probabilistic rough sets, *Inf. Sci.* 180 (3) (2010) 341–353.
- [42] Y.Y. Yao, Probabilistic approaches to rough sets, *Expert Syst.* 20 (5) (2003) 287–297.
- [43] Y.Y. Yao, The superiority of three-way decisions in probabilistic rough set models, *Inf. Sci.* 181 (2011) 1080–1096.
- [44] D.S. Yeung, D.G. Chen, C.C. Tsang, W.T. Lee, X.Z. Wang, On the generalization of fuzzy rough sets, *IEEE Trans. Fuzzy Syst.* 13 (2005) 343–361.
- [45] S.Y. Zhao, C.C. Tsang, D.G. Chen, The model of fuzzy variable precision rough sets, *IEEE Trans. Fuzzy Syst.* 17 (2) (2009) 451–467.
- [46] S.Y. Zhao, C.C. Tsang, D.G. Chen, X.Z. Wang, Building a rule-based classifier by using fuzzy rough set technique, *IEEE Trans. Knowl. Data Eng.* 22 (5) (2010) 624–638.
- [47] S.Y. Zhao, C.C. Tsang, On Fuzzy approximation Operators in Attribute Reduction with Fuzzy Rough Sets, *Inf. Sci.* 178 (16) (2008) 3162–3176.
- [48] W. Zhu, Topological approaches to covering rough sets, *Inf. Sci.* 177 (6) (2007) 1499–1508.
- [49] W. Zhu, F.Y. Wang, On three types of covering-based rough sets, *IEEE Trans. Knowl. Data Eng.* 19 (8) (2007) 1131–1144.
- [50] W. Ziarko, Variable precision rough set model, *J. Comput. Syst. Sci.* 46 (1) (1993) 39–59.
- [51] W. Ziarko, Probabilistic approach to rough sets, *Int. J. Approx. Reason.* 49 (2) (2008) 272–284.
- [52] D.G. Chen, Q.H. Hu, Y.P. Yang, Parameterized attribute reduction with Gaussian kernel based fuzzy rough sets, *Inf. Sci.* 181 (2011) 5169–5179.
- [53] X.Y. Zhang, Z.W. Mo, F. Xiong, W. Cheng, Comparative study of variable precision rough set model and graded rough set model, *Int. J. Approx. Reason.* 53 (2012) 104–116.
- [54] H.Y. Zhang, J. Zhou, D.Q. Miao, C. Gao, Bayesian rough set model: a further investigation, *Int. J. Approx. Reason.* 53 (2012) 541–557.
- [55] T.J. Li, X.P. Yang, An axiomatic characterization of probabilistic rough sets, *Int. J. Approx. Reason.*, in press.
- [56] W. Zhu, F.Y. Wang, The fourth type of covering-based rough sets, *Inf. Sci.* 201 (2012) 80–92.

- [57] H.Y. Zhang, Y. Leung, L. Zhou, Variable-precision-dominance-based rough set approach to interval-valued information systems, *Inf. Sci.* 244 (2013) 75–91.
- [58] N.M. Parthaláin, R. Jensen, Unsupervised fuzzy-rough set-based dimensionality reduction, *Inf. Sci.* 229 (2013) 106–121.
- [59] W. Wei, J.Y. Liang, Y.H. Qian, A comparative study of rough sets for hybrid data, *Inf. Sci.* 190 (2012) 1–16.
- [60] S.M. Vieira, J.M.C. Sousa, U. Kaymak, Fuzzy criteria for feature selection, *Fuzzy Sets Syst.* 189 (2012) 1–18.
- [61] J.H. Dai, H.W. Tian, Fuzzy rough set model for set-valued data, *Fuzzy Sets Syst.*, in press.
- [62] N. Verbiest, C. Cornelis, F. Herrer, FRPS: a fuzzy rough prototype selection method, *Pattern Recog.* 46 (2013) 2770–2782.
- [63] S. Mitra, P.P. Kundu, W. Pedrycz, Feature selection using structural similarity, *Inf. Sci.* 198 (2012) 48–61.