



**QUEEN'S
UNIVERSITY
BELFAST**

A Novel Variable Precision Reduction Approach to Comprehensive Knowledge Systems

Yang, C., Liu, H., McLoone, S., Chen, C. L. P., & Wu, X. (2017). A Novel Variable Precision Reduction Approach to Comprehensive Knowledge Systems. IEEE Transactions on Cybernetics. DOI: 10.1109/TCYB.2017.2648824

Published in:

IEEE Transactions on Cybernetics

Document Version:

Peer reviewed version

Queen's University Belfast - Research Portal:

[Link to publication record in Queen's University Belfast Research Portal](#)

Publisher rights

© 2016 IEEE.

This work is made available online in accordance with the publisher's policies. Please refer to any applicable terms of use of the publisher.

General rights

Copyright for the publications made accessible via the Queen's University Belfast Research Portal is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy

The Research Portal is Queen's institutional repository that provides access to Queen's research output. Every effort has been made to ensure that content in the Research Portal does not infringe any person's rights, or applicable UK laws. If you discover content in the Research Portal that you believe breaches copyright or violates any law, please contact openaccess@qub.ac.uk.

A Novel Variable Precision Reduction Approach to Comprehensive Knowledge Systems

Chao Yang, Hongbo Liu, Sean McLoone, *Senior Member, IEEE*, C. L. Philip Chen, *Fellow, IEEE*,
Xindong Wu, *Fellow, IEEE*

Abstract—A comprehensive knowledge system reveals the intangible insights hidden in an information system by integrating information from multiple data sources in a synthetical manner. In this paper, we present a variable precision reduction theory, underpinned by two new concepts: distribution tables and genealogical binary trees. Sufficient and necessary conditions to extract comprehensive knowledge from a given information system are also presented and proven. A complete variable precision reduction (CVPR) algorithm is proposed, in which we introduce four important strategies, namely, distribution table abstracting, attribute rank dynamic updating, hierarchical binary classifying, and genealogical tree pruning. The completeness of our algorithm is proven theoretically and its superiority to existing methods for obtaining complete reducts is demonstrated experimentally. Finally, having obtaining the complete reduct set, we demonstrate how the relationships between the complete reduct set and comprehensive knowledge system can be visualized in a double-layer lattice structure using Hasse diagrams.

Index Terms—Variable precision reduction, information system, knowledge system, knowledge system structure, data science.

I. INTRODUCTION

DATA science employs theories and techniques drawn from many fields for knowledge extraction from data and information systems [1]. They are usually presented as decision tables with rows and columns, i.e. objects and attributes. Knowledge extraction [2] or rule generation [3] is achieved by reducing the number of attributes in the decision tables in such a way that there is no loss of the information hidden in the information systems. In other words, if there is a metric λ for information, the value of λ is not changed by the reduction process. It is also one of the most common problems

in data mining and knowledge discovery and has been widely studied. Considerable progress has been made on topics such as feature selection [4], [5], dimensionality reduction [6], feature evaluation [7], dynamic updating approximation [8], noisy processing [9], multi-granulation analysis [10], [11] and inconsistent data filtering [12], etc.

When performing knowledge extraction using rough set theory [13], the goal is to obtain a minimum reduct [14], [15] at an affordable computational cost and algorithmic complexity. The corresponding knowledge derived from a single reduct is called a single-knowledge [16]. However, in practice multiple reducts usually exist in information systems, with each reduct having a different attribute combination but the same classification capabilities. These can be thought of as providing insights from different perspectives [17]. Multiple reducts can constitute a multi-knowledge system [18], but this does not guarantee that they capture all available knowledge. For a given information system, only a complete reduct set contains all possible reducts without redundancy, and therefore can be used to deduce a comprehensive knowledge system. Unfortunately, obtaining a complete reduct set is an NP-hard problem [19], and as such represents a major challenge computationally.

This paper focuses on how to obtain the complete reducts set, formulate the corresponding knowledge structure and generate comprehensive knowledge systems. The paper has the following contributions:

- We present a variable precision reduction theory for comprehensive knowledge systems. Some underpinning concepts relating to the completeness of reduct set and knowledge are defined. In particular, we introduce two new concepts, namely, distribution tables and genealogical binary trees, and establish the necessary and sufficient conditions to achieve complete multi-knowledge extraction without redundancies.
- An efficient and effective complete variable precision reduction algorithm is then proposed, the key elements of which are distribution table abstracting, attribute rank dynamic updating, hierarchical binary classifying and genealogical tree pruning. We prove theoretically the completeness of our algorithm and compare its efficacy to existing methods with the aid of experimental studies.
- We also explore the relationships between a complete reduct set and its knowledge system, introducing a double-layer lattice structure to enable them to be illustrated visually in Hasse diagrams.

Manuscript accepted 2 December 2016. This work is supported partly by the Macau Science and Technology Development under Grant 019/2015/A and Multiyear Research Grants, the National Natural Science Foundation of China (Grant No. 61173035, 61472058, 61572540), the Program for New Century Excellent Talents in University (Grant No. NCET-11-0861) and the Program for Changjiang Scholars and Innovative Research Team in University (PCSIRT) of the Ministry of Education, China (Grant No. IRT13059).

C. Yang and H. Liu are with the School of Science and Technology at Dalian Maritime University, Dalian, China (e-mail: yangchaoneu@sina.com, lhb@dlmu.edu.cn).

S. McLoone is with the School of Electronics, Electrical Engineering and Computer Science at Queen's University Belfast, Northern Ireland, UK (e-mail: s.mcloone@qub.ac.uk).

C. L. Philip Chen is with the Faculty of Science and Technology, University of Macau, Macau, China, UMacau Research Institute, Zhuhai, Guangdong, China and School of Information Science and Technology at Dalian University of Technology, Dalian 116023, China (e-mail: Philip.Chen@ieee.org).

X. Wu is with the School of Computer Science and Information Engineering, Hefei University of Technology, Hefei, China, and the Department of Computer Science, University of Vermont, USA (e-mail: xwu@uvm.edu)

The remainder of the paper is organized as follows. Related work is reviewed in Section II. In Sections III we present our variable precision reduction theory to include the completeness of reduct set and knowledge extraction. In Section IV, we propose a complete variable precision reduction algorithm. In particular, the dynamic attribute ranking to a genealogical binary tree is presented in Subsection IV-A, while the hierarchical binary classifying and pruning strategies are introduced in Subsection IV-B. Then, our complete reduction and knowledge extraction are presented in Subsection IV-C. The experimental results and discussions are provided in Section V. Finally, conclusions and areas for future work are given in Section VI.

II. RELATED WORK

A. Knowledge Comprehensiveness

Pancerz and his colleagues discussed information and dynamic information systems, which can be used in prediction problems [20], [21]. Compared to single-knowledge, multi-knowledge is usually more comprehensive, and as such is likely to correctly classify new objects. Abu-Donia [22] presented rough set approximations using multi-knowledge base to raise the efficiency of decision support system. Gams [23] demonstrated the creation of a multi-knowledge system by integrating ten single-knowledge systems using 10-fold partition learning. They illustrated that a multi-knowledge system can improve the accuracy of classification significantly. However, although they put the multi-knowledge system into practice, issues of redundancy and comprehensiveness of the knowledge extracted were not considered. Ma [24] introduced a completeness condition to determine whether an attribute combination is a reduct or not, but again does not consider the completeness of multi-reducts. A complete reduct set is stricter than multi-reducts, since in a complete reduct set, all reducts are included without redundancy. Knowledge comprehensiveness also depends on the corresponding complete reduct set, because different reduct provide knowledge from different perspectives.

B. Knowledge Visualisation

Qian et al. presented a space distance [25] which is used to characterize the similarity between reducts. After attribute reduction, a knowledge system can be constituted by rule extraction tools, such as rough set [26], apriori algorithm [27], formal concept analysis [28] and so on. Formal concept analysis not only gives implications " $X \rightarrow Y$ ", but also builds a hierarchical structure of knowledge concepts through concept lattices [29] which can be used to describe the relationships between objects and attributes. Through constructing complete concept lattices, the lattice structure can be introduced to describe the similarity between reducts or between knowledge, and the relationship between a reduct and its corresponding knowledge can also be depicted. Hasse diagrams [30] are a suitable way to visualise the lattices of the comprehensive knowledge system structure as developed in this paper.

C. Knowledge Extraction Approaches

Different heuristic algorithms have been proposed to reduce attributes and to extract knowledge from information systems in the classical rough set model [31]. However, these algorithms are subject to a rigid premise in the classical rough set model that the error classification rate [32] must be zero.

To relax this rigid premise of the classical rough set model, variable precision rough sets (VPRS) [33] are presented to introduce an error classification rate β [34]. The positive and boundary regions in rough set theory are designed to distinguish certain and uncertain objects, respectively [35], [36]. The predictable (certain) objects will increase by reducing β .

Ziarko presents a β -reduct method [33] for attribute reduction while Park and Choi propose a novel VPRS method using information entropy [37]. However, a side effect of their approaches is the so called "jumping" phenomenon [14], [38], where the output alternates between reduct and non-reduct.

Wang et al. [38] show that the "jumping" phenomenon is due to the fact that the changes in classification quality and positive region are non-monotonous. They further illustrate that the distribution of the lower approximation bound about decision classes varies monotonically. The reduction approach in this case follows a decision monotonicity criterion [39], [40]. It implies that rough set reduction is subject to the distribution of the objects with β -dominant decision value in each equivalence class to β extent within the positive region, while the distribution of other objects have little influence because of the error classification rate β .

The attribute reduction methods based on the discernibility matrix and the ones based on the positive region can both be used to obtain multiple reducts from a given information system. However, the former is usually much more computationally intensive and is limited to much lower dimension problems than the later, even when using the approach introduced by Yang [41] to minimize the number of elements in the discernibility matrix to decrease the computational load. In the case of multi-reducts and multi-knowledge extraction methods based on the positive region, the related work mainly falls into two categories:

- non-core attribute combination algorithms, such as the worst-attribute-drop-first (WADF) algorithm [16]. In these algorithms, all attributes are divided into core attributes and non-core attributes. A reduct is obtained and then used as a seed reduct to generate other reducts in the multi-reducts set through a non-core attribute replacement process, whereby one non-core attribute is excluded in the identified reduct. In [42] a multi-knowledge system is constituted to help robots identify their environment. However, in the algorithms employed, the number of multi-reducts obtained is strongly dependent on the initial seed reducts.
- non-deterministic random or quasi-random search algorithm, such as swarm intelligent algorithms [43], [44]. These algorithms utilize a swarm of intelligent individuals to search for the reducts in an information system and can usually obtain multi-reducts quickly. They have been employed to obtain a multi-knowledge system using

rough sets and to complete multi-factor analysis of violent crime [45]. However, although random search algorithms can converge theoretically with a probability of 1, they cannot guarantee the completeness of multi-reducts within a finite run time in practice.

III. REDUCTION AND KNOWLEDGE THEORY

The basic concepts of attribute reduction and its philosophy are presented in the related literature [38], [46] and multiple reducts and multi-knowledge systems are described in [17]. Here, we will recall the reduction concepts that are relevant to the present work and then extend the relevant theories and concepts to provide a theoretical description of comprehensive knowledge systems.

A. Distribution Table

An information system can be defined as a decision table by $T = (U, C, D, V, f)$, where C is the set of condition attributes and D is the set of decision attributes, respectively. V is the value set of all attributes. $f : U \times \{C \cup D\} \rightarrow V$ is the information function such that $f(x, a) \in V_a$ for every $a \in C \cup D$, $x \in U$. For $E \subseteq C \cup D$, an equivalence relation $IND(E)$ is defined as follows:

$$IND(E) = \{(x, y) \in U \times U \mid \forall a \in E, f(x, a) = f(y, a)\} \quad (1)$$

$IND(E)$ partitions U into disjoint subsets. Let U/E denote the family of all equivalence classes of the relation $IND(E)$, i.e. $U/E = \{E_1, E_2, \dots, E_i, \dots\}$, where E_i is an equivalence class of E , which is denoted $[x_i]_E$. Note that equivalence classes are defined with respect to their own attribute set. Equivalence classes U/C and U/D will be called condition and decision classes, respectively.

Definition 1: [Positive region] Given a decision table $T = (U, C, D, V, f)$. Let $B \subseteq C$. The B -positive region with β of D is the set of all objects from U which satisfy Equ. (2).

$$POS_B^\beta(D) = \bigcup_{\frac{|[x_i]_B \cap [x_i]_D|}{|[x_i]_B|} \geq \beta} [x_i]_B \quad (2)$$

where $\beta \in (0.5, 1]$. In particular, the positive region with β will degenerate into the classical rough set model if $\beta = 1$.

Definition 2: [β -decision table] Given a decision table $T = (U, C, D, V, f)$, determine its condition classes $[x_i]_C$. The decision value of objects in each condition class to β extent within $POS_C^\beta(D)$ is normalized to the β -dominant decision value in the respective condition class. Namely, the decision value of $[x_i]_C$ in the positive region is normalized such that:

$$\frac{|[x_i]_C \cap [x_i]_D|}{|[x_i]_C|} = 1 \quad (3)$$

and then, the information function is updated as $f_\beta : U \times \{C \cup D\} \rightarrow V_\beta$. In this way a β -decision table is formulated, which will be denoted as $T_\beta = (U, C, D, V_\beta, f_\beta)$. Accordingly, attribute $b \in B \subseteq C$ is D -dispensable in B , if $POS_B^\beta(D) = POS_{B-\{b\}}^\beta(D)$; otherwise attribute b is D -indispensable in B . If $\forall b \in B$ are D -indispensable in B , then B will be called D -independent. A subset of attributes $B \subseteq C$ is

a D -reduct of C , iff $POS_B^\beta(D) = POS_C^\beta(D)$ and B is D -independent.

Theorem 1: Given T_β , set $NEG_C^\beta(D) = U - POS_C^\beta(D)$. $B \subseteq C$ is a reduct, therefore $\forall [x_i]_B$ either $[x_i]_B \subseteq NEG_C^\beta(D)$ or $[x_i]_B \subseteq POS_C^\beta(D) \wedge |[x_i]_B/D| = 1$ is satisfied.

Proof 1: B is a reduct, so $POS_C^\beta(D) = POS_B^\beta(D)$ and $NEG_C^\beta(D) = NEG_B^\beta(D)$. If $[x_i]_B \subseteq NEG_B^\beta(D)$, $[x_i]_B \subseteq NEG_C^\beta(D)$. Similarly, $[x_i]_B \subseteq POS_C^\beta(D)$. According to

Definition 2, $\frac{|[x_i]_B \cap [x_i]_D|}{|[x_i]_B|} = 1 \Leftrightarrow |[x_i]_B/D| = 1$.

Definition 3: [Distribution table] Given a decision table $T = (U, C, D, V, f)$, transform it into β -decision table $T_\beta = (U, C, D, V_\beta, f_\beta)$, in which the decision value of x_i according to a decision monotonicity criterion is adjusted by Equ. (4).

$$f_\beta(x_i, D) = \begin{cases} \arg \max_D \left\{ \frac{|[x_i]_C \cap [x_i]_D|}{|[x_i]_C|} \right\} & \text{if } \beta \leq \rho(x_i) \leq 1, \\ f(x_i, D) & \text{otherwise.} \end{cases} \quad (4)$$

where $\rho(x_i) = \frac{|[x_i]_C \cap [x_i]_D|}{|[x_i]_C|}$. $U/C = \{[x_1]_C, \dots, [x_i]_C, \dots, [x_m]_C\}$ is the equivalence class set. Here a distribution table $T_D = (U_D(\omega), C, D, V_\beta, f_\beta)$ can be formulated with the same attributes while $U_D = \{(x_1, \omega(x_1)), \dots, (x_i, \omega(x_i)), \dots, (x_m, \omega(x_m))\}$, where the distribution region label $\omega(x_i)$ of object x_i is determined by Equ. (5).

$$\omega(x_i) = \begin{cases} 1 & \text{if } \beta \leq \rho(x_i) \leq 1, \\ 0 & \text{otherwise.} \end{cases} \quad (5)$$

These objects are in the positive region if $\omega(x_i) = 1$ and in the negative region if $\omega(x_i) = 0$. In other words, all objects have their deterministic distributions during variable precision reduction, which helps us avoid the “jumping” phenomenon. Furthermore, the distribution table collects only one object from each equivalence class in a given information system. It is an essential abstract from the original universe of discourse, and reduces significantly the number of considered objects, especially for large datasets. These benefits motivate our use of the distribution table instead of working directly with the decision table. We will prove the equivalence between the distribution table and the original decision table in Theorem 2 for attribute reduction.

Theorem 2: Given T , compute T_D , where $U_D = \{(x_1, \omega(x_1)), \dots, (x_i, \omega(x_i)), \dots, (x_m, \omega(x_m))\}$. $\forall B \subseteq C$, if an arbitrary equivalence class $[x_i]_B$ in T_D satisfies $\omega(x_i) = 0$ or $\omega(x_i) = 1 \wedge |[x_i]_B/D| = 1$ for $\forall x_i \in [x_i]_B$, then $POS_B^\beta(D) = POS_C^\beta(D)$.

Proof 2: $[x_i]_B$ respects an arbitrary equivalence class about x_i in T . By Definition 3, $\forall x_i \in [x_i]_B$, $\omega(x_i) = 0 \Leftrightarrow [x_i]_B \subseteq NEG_C^\beta(D) \Leftrightarrow [x_i]_B \subseteq NEG_C^\beta(D)$.

By Theorem 1 ($\forall x_i \in [x_i]_B$, $\omega(x_i) = 1$ and $|[x_i]_B/D| = 1$) $\Leftrightarrow ([x_i]_B \subseteq POS_C^\beta(D) \text{ and } \beta > \frac{1}{2}) \Leftrightarrow [x_i]_B \subseteq POS_C^\beta(D)$. Thus, $POS_B^\beta(D) = POS_C^\beta(D)$.

B. Reduct Completeness

Definition 4: [Complete reduct set] Consider all possible attribute subsets $\{\{b_1\}, \dots, \{b_{|C|}\}, \{b_1, b_2\}, \dots, \{b_1, \dots, b_{|C|}\}\}$, where $b_i \in B \subseteq C$, $1 \leq i \leq |C|$. Let $RED(\beta)$ represent the set of complete reducts, i.e.

$$RED(\beta) = \{B | POS_B^\beta(D) = POS_C^\beta(D), \forall b \in B, POS_{(B-\{b\})}^\beta(D) < POS_B^\beta(D)\}. \quad (6)$$

Definition 5: [Reduct completeness ratio] The reduct completeness ratio is defined as follows.

$$\vartheta = \frac{|RED(\beta)|}{|RED(\hat{\beta})|} \quad (7)$$

where $RED(\hat{\beta})$ represents the set of obtained reducts in practice.

Theorem 3: Given a β -decision table $T_\beta = (U, C, D, V_\beta, f_\beta)$, $B \subseteq C$, then $x \in POS_B^\beta(D_\beta)$ iff x satisfies the predicate formulas $(\forall y \in U)(D_\beta(x) \neq D_\beta(y) \Rightarrow B(x) \neq B(y))$ is true where $D_\beta(x)$ represents the decision value of x in T_β .

Proof 3:

$$\begin{aligned} x \in POS_B^\beta(D_\beta) &\Leftrightarrow [x]_B \subseteq [x]_{D_\beta} \\ &\Leftrightarrow (\forall y \in U)(B(x) = B(y) \Rightarrow D_\beta(x) = D_\beta(y)) \\ &\Leftrightarrow (\forall y \in U)(B(x) \neq B(y) \vee D_\beta(x) = D_\beta(y)) \\ &\Leftrightarrow (\forall y \in U)(D_\beta(x) = D_\beta(y) \vee B(x) = B(y)) \\ &\Leftrightarrow (\forall y \in U)(D_\beta(x) \neq D_\beta(y) \Rightarrow B(x) \neq B(y)). \end{aligned}$$

Theorem 4: $POS_B^\beta(D_\beta) \subseteq POS_C^\beta(D_\beta)$ if $B \subseteq C$.

Proof 4: Set $x \in POS_B^\beta(D_\beta)$, we have $[x]_B \subseteq [x]_{D_\beta}$. Since $B \subseteq C$, $[x]_C \subseteq [x]_B$. Therefore $[x]_C \subseteq [x]_{D_\beta}$. We have $x \in POS_C^\beta(D_\beta)$.

Theorem 5: Set $B \subseteq C$, $POS_B^\beta(D_\beta) = POS_C^\beta(D_\beta)$ iff B satisfies the following predicate formulas:

$$\begin{aligned} &(\forall x \in U)((\exists y \in U)(D_\beta(x) \neq D_\beta(y) \wedge C(x) = C(y)) \\ &\vee (\forall z \in U)((D_\beta(x) \neq D_\beta(z) \Rightarrow B(x) \neq B(z))). \end{aligned}$$

Proof 5: Since $B \subseteq C$, we have $POS_B^\beta(D_\beta) \subseteq POS_C^\beta(D_\beta)$ from Theorem 4. So $(POS_B^\beta(D_\beta) = POS_C^\beta(D_\beta)) \Leftrightarrow (POS_B^\beta(D_\beta) \supseteq POS_C^\beta(D_\beta))$, that is $(POS_B^\beta(D_\beta) = POS_C^\beta(D_\beta)) \Leftrightarrow ((\forall x \in U)(x \in POS_C^\beta(D_\beta) \Rightarrow x \in POS_B^\beta(D_\beta)))$.

From Theorem 3, we have

$$\begin{aligned} POS_B^\beta(D_\beta) = POS_C^\beta(D_\beta) &\Leftrightarrow \\ &((\forall x \in U)((\forall y \in U)(D_\beta(x) \neq D_\beta(y) \Rightarrow C(x) \neq C(y)) \\ &\Rightarrow ((\forall z \in U)(D_\beta(x) \neq D_\beta(z) \Rightarrow B(x) \neq B(z)))) \Leftrightarrow \\ &((\forall x \in U)(\forall y \in U)(D_\beta(x) \neq D_\beta(y) \Rightarrow C(x) \neq C(y)) \\ &\vee (\forall z \in U)(D_\beta(x) \neq D_\beta(z) \Rightarrow B(x) \neq B(z))) \Leftrightarrow \\ &((\forall x \in U)((\exists y \in U)(D_\beta(x) \neq D_\beta(y) \wedge C(x) = C(y)) \\ &\vee (\forall z \in U)(D_\beta(x) \neq D_\beta(z) \Rightarrow B(x) \neq B(z))))). \end{aligned}$$

When $\beta = 1$, Theorem 5 is consistent with the results in [24].

Theorem 6: $\forall B \subseteq C$, if $[x_i]_C/D \neq 1$, $[x_i]_B/D \neq 1$.

Proof 6: Because $B \subseteq C$, $[x_i]_C \subseteq [x_i]_B$. And $[x_i]_C/D \neq 1$, it must be $[x_i]_B/D \neq 1$.

C. Genealogical Binary Tree

In a left-child-right-sibling (LCRS) binary tree $M = (N, L, R)$, N denotes a node, its parent node N_f and sibling node N_s , L and R is the left and right child nodes, respectively. There is no child node in a leaf node.

Definition 6: [Brotherhood] In the right path of the LCRS binary tree, the previous node of the starting node is null or N_f and the terminator node is a leaf node, then all of nodes in this path is a brotherhood in the binary tree.

In particular, the brotherhood is also called the ancestor when the previous node of the starting node is null.

Definition 7: [Genealogical binary tree] Given $T = (U, C, D, V, f)$, a LCRS binary tree maps C to generate a genealogical binary tree, denoted as $G(C)$, where any attribute c in C is mapped into a node \odot . In the right path of $G(C)$, the previous nodes of node \odot are its elder siblings, while those next nodes of node \odot are its younger siblings. In the left path of $G(C)$, the path from a node to its ancestor is called as a genealogical generation path, denoted as " \curvearrowright ". The nodes in a genealogical generation path only have parent-child relationships without brotherhood. By traversing all genealogical generation paths the complete attribute combinations set, denoted $P^{G(C)}$, can be obtained.

The genealogical binary tree of T will provide a deterministic roadmap to perform an orderly and complete search of all candidate reducts in T .

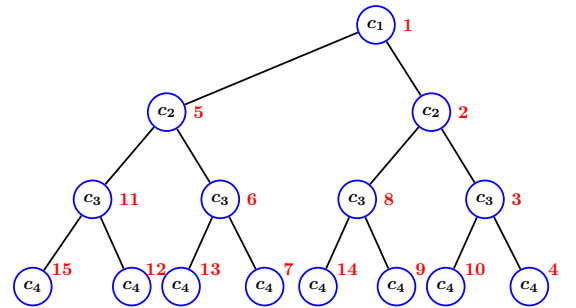


Fig. 1: A genealogical binary tree

As shown in Figure 1, a genealogical binary tree is mapped from $C = \{c_1, c_2, c_3, c_4\}$ in a given $T = (U, C, D, V, f)$. The number at the upper right of each node is its sequence number. Here, the order of the attributes in T is used as the order of the attributes in the nodes. In practice, the attributes need to be ranked before attribute reduction. We will discuss the ranking process later (see Equ. (15)).

Nodes 1, 2, 3 and 4 are the brotherhood (ancestor). Nodes 1 and 2 are the elder siblings of node 3 and its younger sibling is node 4. In the left paths of $G(C)$, all attribute combinations can be obtained from $P^{G(C)}$, i.e. $P^{G(C)} = \{\{c_1\}, \{c_2\}, \{c_3\}, \{c_4\}, \{c_1, c_2\}, \{c_1, c_3\}, \{c_1, c_4\}, \{c_2, c_3\}, \{c_2, c_4\}, \{c_3, c_4\}, \{c_1, c_2, c_3\}, \{c_1, c_2, c_4\}, \{c_1, c_3, c_4\}, \{c_2, c_3, c_4\}, \{c_1, c_2, c_3, c_4\}\}$. The corresponding genealogical generation paths are ①, ②, ③, ④, ① \curvearrowright ⑤, ① \curvearrowright ⑥, ① \curvearrowright ⑦, ② \curvearrowright ⑧, ② \curvearrowright ⑨, ③ \curvearrowright ⑩, ③ \curvearrowright ⑪, ③ \curvearrowright ⑫, ③ \curvearrowright ⑬, ③ \curvearrowright ⑭, ③ \curvearrowright ⑮, ③ \curvearrowright ⑯, ③ \curvearrowright ⑰, ③ \curvearrowright ⑱, ③ \curvearrowright ⑲, ③ \curvearrowright ⑳, ③ \curvearrowright ㉑, ③ \curvearrowright ㉒, ③ \curvearrowright ㉓, ③ \curvearrowright ㉔, ③ \curvearrowright ㉕, ③ \curvearrowright ㉖, ③ \curvearrowright ㉗, ③ \curvearrowright ㉘, ③ \curvearrowright ㉙, ③ \curvearrowright ㉚, ③ \curvearrowright ㉛, ③ \curvearrowright ㉜, ③ \curvearrowright ㉝, ③ \curvearrowright ㉞, ③ \curvearrowright ㉟, ③ \curvearrowright ㊱, ③ \curvearrowright ㊲, ③ \curvearrowright ㊳, ③ \curvearrowright ㊴, ③ \curvearrowright ㊵, ③ \curvearrowright ㊶, ③ \curvearrowright ㊷, ③ \curvearrowright ㊸, ③ \curvearrowright ㊹, ③ \curvearrowright ㊺, ③ \curvearrowright ㊻, ③ \curvearrowright ㊼, ③ \curvearrowright ㊽, ③ \curvearrowright ㊾, ③ \curvearrowright ㊿, ③ \curvearrowright ㏀, ③ \curvearrowright ㏁, ③ \curvearrowright ㏂, ③ \curvearrowright ㏃, ③ \curvearrowright ㏄, ③ \curvearrowright ㏅, ③ \curvearrowright ㏆, ③ \curvearrowright ㏇, ③ \curvearrowright ㏈, ③ \curvearrowright ㏉, ③ \curvearrowright ㏊, ③ \curvearrowright ㏋, ③ \curvearrowright ㏌, ③ \curvearrowright ㏍, ③ \curvearrowright ㏎, ③ \curvearrowright ㏏, ③ \curvearrowright ㏐, ③ \curvearrowright ㏑, ③ \curvearrowright ㏒, ③ \curvearrowright ㏓, ③ \curvearrowright ㏔, ③ \curvearrowright ㏕, ③ \curvearrowright ㏖, ③ \curvearrowright ㏗, ③ \curvearrowright ㏘, ③ \curvearrowright ㏙, ③ \curvearrowright ㏚, ③ \curvearrowright ㏛, ③ \curvearrowright ㏜, ③ \curvearrowright ㏝, ③ \curvearrowright ㏞, ③ \curvearrowright ㏟, ③ \curvearrowright ㏠, ③ \curvearrowright ㏡, ③ \curvearrowright ㏢, ③ \curvearrowright ㏣, ③ \curvearrowright ㏤, ③ \curvearrowright ㏥, ③ \curvearrowright ㏦, ③ \curvearrowright ㏧, ③ \curvearrowright ㏨, ③ \curvearrowright ㏩, ③ \curvearrowright ㏪, ③ \curvearrowright ㏫, ③ \curvearrowright ㏬, ③ \curvearrowright ㏭, ③ \curvearrowright ㏮, ③ \curvearrowright ㏯, ③ \curvearrowright ㏰, ③ \curvearrowright ㏱, ③ \curvearrowright ㏲, ③ \curvearrowright ㏳, ③ \curvearrowright ㏴, ③ \curvearrowright ㏵, ③ \curvearrowright ㏶, ③ \curvearrowright ㏷, ③ \curvearrowright ㏸, ③ \curvearrowright ㏹, ③ \curvearrowright ㏺, ③ \curvearrowright ㏻, ③ \curvearrowright ㏼, ③ \curvearrowright ㏽, ③ \curvearrowright ㏾, ③ \curvearrowright ㏿, ③ \curvearrowright 㐀, ③ \curvearrowright 㐁, ③ \curvearrowright 㐂, ③ \curvearrowright 㐃, ③ \curvearrowright 㐄, ③ \curvearrowright 㐅, ③ \curvearrowright 㐆, ③ \curvearrowright 㐇, ③ \curvearrowright 㐈, ③ \curvearrowright 㐉, ③ \curvearrowright 㐊, ③ \curvearrowright 㐋, ③ \curvearrowright 㐌, ③ \curvearrowright 㐍, ③ \curvearrowright 㐎, ③ \curvearrowright 㐏, ③ \curvearrowright 㐐, ③ \curvearrowright 㐑, ③ \curvearrowright 㐒, ③ \curvearrowright 㐓, ③ \curvearrowright 㐔, ③ \curvearrowright 㐕, ③ \curvearrowright 㐖, ③ \curvearrowright 㐗, ③ \curvearrowright 㐘, ③ \curvearrowright 㐙, ③ \curvearrowright 㐚, ③ \curvearrowright 㐛, ③ \curvearrowright 㐜, ③ \curvearrowright 㐝, ③ \curvearrowright 㐞, ③ \curvearrowright 㐟, ③ \curvearrowright 㐠, ③ \curvearrowright 㐡, ③ \curvearrowright 㐢, ③ \curvearrowright 㐣, ③ \curvearrowright 㐤, ③ \curvearrowright 㐥, ③ \curvearrowright 㐦, ③ \curvearrowright 㐧, ③ \curvearrowright 㐨, ③ \curvearrowright 㐩, ③ \curvearrowright 㐪, ③ \curvearrowright 㐫, ③ \curvearrowright 㐬, ③ \curvearrowright 㐭, ③ \curvearrowright 㐮, ③ \curvearrowright 㐯, ③ \curvearrowright 㐰, ③ \curvearrowright 㐱, ③ \curvearrowright 㐲, ③ \curvearrowright 㐳, ③ \curvearrowright 㐴, ③ \curvearrowright 㐵, ③ \curvearrowright 㐶, ③ \curvearrowright 㐷, ③ \curvearrowright 㐸, ③ \curvearrowright 㐹, ③ \curvearrowright 㐺, ③ \curvearrowright 㐻, ③ \curvearrowright 㐼, ③ \curvearrowright 㐽, ③ \curvearrowright 㐾, ③ \curvearrowright 㐿, ③ \curvearrowright 㑀, ③ \curvearrowright 㑁, ③ \curvearrowright 㑂, ③ \curvearrowright 㑃, ③ \curvearrowright 㑄, ③ \curvearrowright 㑅, ③ \curvearrowright 㑆, ③ \curvearrowright 㑇, ③ \curvearrowright 㑈, ③ \curvearrowright 㑉, ③ \curvearrowright 㑊, ③ \curvearrowright 㑋, ③ \curvearrowright 㑌, ③ \curvearrowright 㑍, ③ \curvearrowright 㑎, ③ \curvearrowright 㑏, ③ \curvearrowright 㑐, ③ \curvearrowright 㑑, ③ \curvearrowright 㑒, ③ \curvearrowright 㑓, ③ \curvearrowright 㑔, ③ \curvearrowright 㑕, ③ \curvearrowright 㑖, ③ \curvearrowright 㑗, ③ \curvearrowright 㑘, ③ \curvearrowright 㑙, ③ \curvearrowright 㑚, ③ \curvearrowright 㑛, ③ \curvearrowright 㑜, ③ \curvearrowright 㑝, ③ \curvearrowright 㑞, ③ \curvearrowright 㑟, ③ \curvearrowright 㑠, ③ \curvearrowright 㑡, ③ \curvearrowright 㑢, ③ \curvearrowright 㑣, ③ \curvearrowright 㑤, ③ \curvearrowright 㑥, ③ \curvearrowright 㑦, ③ \curvearrowright 㑧, ③ \curvearrowright 㑨, ③ \curvearrowright 㑩, ③ \curvearrowright 㑪, ③ \curvearrowright 㑫, ③ \curvearrowright 㑬, ③ \curvearrowright 㑭, ③ \curvearrowright 㑮, ③ \curvearrowright 㑯, ③ \curvearrowright 㑰, ③ \curvearrowright 㑱, ③ \curvearrowright 㑲, ③ \curvearrowright 㑳, ③ \curvearrowright 㑴, ③ \curvearrowright 㑵, ③ \curvearrowright 㑶, ③ \curvearrowright 㑷, ③ \curvearrowright 㑸, ③ \curvearrowright 㑹, ③ \curvearrowright 㑺, ③ \curvearrowright 㑻, ③ \curvearrowright 㑼, ③ \curvearrowright 㑽, ③ \curvearrowright 㑾, ③ \curvearrowright 㑿, ③ \curvearrowright 㒀, ③ \curvearrowright 㒁, ③ \curvearrowright 㒂, ③ \curvearrowright 㒃, ③ \curvearrowright 㒄, ③ \curvearrowright 㒅, ③ \curvearrowright 㒆, ③ \curvearrowright 㒇, ③ \curvearrowright 㒈, ③ \curvearrowright 㒉, ③ \curvearrowright 㒊, ③ \curvearrowright 㒋, ③ \curvearrowright 㒌, ③ \curvearrowright 㒍, ③ \curvearrowright 㒎, ③ \curvearrowright 㒏, ③ \curvearrowright 㒐, ③ \curvearrowright 㒑, ③ \curvearrowright 㒒, ③ \curvearrowright 㒓, ③ \curvearrowright 㒔, ③ \curvearrowright 㒕, ③ \curvearrowright 㒖, ③ \curvearrowright 㒗, ③ \curvearrowright 㒘, ③ \curvearrowright 㒙, ③ \curvearrowright 㒚, ③ \curvearrowright 㒛, ③ \curvearrowright 㒜, ③ \curvearrowright 㒝, ③ \curvearrowright 㒞, ③ \curvearrowright 㒟, ③ \curvearrowright 㒠, ③ \curvearrowright 㒡, ③ \curvearrowright 㒢, ③ \curvearrowright 㒣, ③ \curvearrowright 㒤, ③ \curvearrowright 㒥, ③ \curvearrowright 㒦, ③ \curvearrowright 㒧, ③ \curvearrowright 㒨, ③ \curvearrowright 㒩, ③ \curvearrowright 㒪, ③ \curvearrowright 㒫, ③ \curvearrowright 㒬, ③ \curvearrowright 㒭, ③ \curvearrowright 㒮, ③ \curvearrowright 㒯, ③ \curvearrowright 㒰, ③ \curvearrowright 㒱, ③ \curvearrowright 㒲, ③ \curvearrowright 㒳, ③ \curvearrowright 㒴, ③ \curvearrowright 㒵, ③ \curvearrowright 㒶, ③ \curvearrowright 㒷, ③ \curvearrowright 㒸, ③ \curvearrowright 㒹, ③ \curvearrowright 㒺, ③ \curvearrowright 㒻, ③ \curvearrowright 㒼, ③ \curvearrowright 㒽, ③ \curvearrowright 㒾, ③ \curvearrowright 㒿, ③ \curvearrowright 㓀, ③ \curvearrowright 㓁, ③ \curvearrowright 㓂, ③ \curvearrowright 㓃, ③ \curvearrowright 㓄, ③ \curvearrowright 㓅, ③ \curvearrowright 㓆, ③ \curvearrowright 㓇, ③ \curvearrowright 㓈, ③ \curvearrowright 㓉, ③ \curvearrowright 㓊, ③ \curvearrowright 㓋, ③ \curvearrowright 㓌, ③ \curvearrowright 㓍, ③ \curvearrowright 㓎, ③ \curvearrowright 㓏, ③ \curvearrowright 㓐, ③ \curvearrowright 㓑, ③ \curvearrowright 㓒, ③ \curvearrowright 㓓, ③ \curvearrowright 㓔, ③ \curvearrowright 㓕, ③ \curvearrowright 㓖, ③ \curvearrowright 㓗, ③ \curvearrowright 㓘, ③ \curvearrowright 㓙, ③ \curvearrowright 㓚, ③ \curvearrowright 㓛, ③ \curvearrowright 㓜, ③ \curvearrowright 㓝, ③ \curvearrowright 㓞, ③ \curvearrowright 㓟, ③ \curvearrowright 㓠, ③ \curvearrowright 㓡, ③ \curvearrowright 㓢, ③ \curvearrowright 㓣, ③ \curvearrowright 㓤, ③ \curvearrowright 㓥, ③ \curvearrowright 㓦, ③ \curvearrowright 㓧, ③ \curvearrowright 㓨, ③ \curvearrowright 㓩, ③ \curvearrowright 㓪, ③ \curvearrowright 㓫, ③ \curvearrowright 㓬, ③ \curvearrowright 㓭, ③ \curvearrowright 㓮, ③ \curvearrowright 㓯, ③ \curvearrowright 㓰, ③ \curvearrowright 㓱, ③ \curvearrowright 㓲, ③ \curvearrowright 㓳, ③ \curvearrowright 㓴, ③ \curvearrowright 㓵, ③ \curvearrowright 㓶, ③ \curvearrowright 㓷, ③ \curvearrowright 㓸, ③ \curvearrowright 㓹, ③ \curvearrowright 㓺, ③ \curvearrowright 㓻, ③ \curvearrowright 㓼, ③ \curvearrowright 㓽, ③ \curvearrowright 㓾, ③ \curvearrowright 㓿, ③ \curvearrowright 㔀, ③ \curvearrowright 㔁, ③ \curvearrowright 㔂, ③ \curvearrowright 㔃, ③ \curvearrowright 㔄, ③ \curvearrowright 㔅, ③ \curvearrowright 㔆, ③ \curvearrowright 㔇, ③ \curvearrowright 㔈, ③ \curvearrowright 㔉, ③ \curvearrowright 㔊, ③ \curvearrowright 㔋, ③ \curvearrowright 㔌, ③ \curvearrowright 㔍, ③ \curvearrowright 㔎, ③ \curvearrowright 㔏, ③ \curvearrowright 㔐, ③ \curvearrowright 㔑, ③ \curvearrowright 㔒, ③ \curvearrowright 㔓, ③ \curvearrowright 㔔, ③ \curvearrowright 㔕, ③ \curvearrowright 㔖, ③ \curvearrowright 㔗, ③ \curvearrowright 㔘, ③ \curvearrowright 㔙, ③ \curvearrowright 㔚, ③ \curvearrowright 㔛, ③ \curvearrowright 㔜, ③ \curvearrowright 㔝, ③ \curvearrowright 㔞, ③ \curvearrowright 㔟, ③ \curvearrowright 㔠, ③ \curvearrowright 㔡, ③ \curvearrowright 㔢, ③ \curvearrowright 㔣, ③ \curvearrowright 㔤, ③ \curvearrowright 㔥, ③ \curvearrowright 㔦, ③ \curvearrowright 㔧, ③ \curvearrowright 㔨, ③ \curvearrowright 㔩, ③ \curvearrowright 㔪, ③ \curvearrowright 㔫, ③ \curvearrowright 㔬, ③ \curvearrowright 㔭, ③ \curvearrowright 㔮, ③ \curvearrowright 㔯, ③ \curvearrowright 㔰, ③ \curvearrowright 㔱, ③ \curvearrowright 㔲, ③ \curvearrowright 㔳, ③ \curvearrowright 㔴, ③ \curvearrowright 㔵, ③ \curvearrowright 㔶, ③ \curvearrowright 㔷, ③ \curvearrowright 㔸, ③ \curvearrowright 㔹, ③ \curvearrowright 㔺, ③ \curvearrowright 㔻, ③ \curvearrowright 㔼, ③ \curvearrowright 㔽, ③ \curvearrowright 㔾, ③ \curvearrowright 㔿, ③ \curvearrowright 㕀, ③ \curvearrowright 㕁, ③ \curvearrowright 㕂, ③ \curvearrowright 㕃, ③ \curvearrowright 㕄, ③ \curvearrowright 㕅, ③ \curvearrowright 㕆, ③ \curvearrowright 㕇, ③ \curvearrowright 㕈, ③ \curvearrowright 㕉, ③ \curvearrowright 㕊, ③ \curvearrowright 㕋, ③ \curvearrowright 㕌, ③ \curvearrowright 㕍, ③ \curvearrowright 㕎, ③ \curvearrowright 㕏, ③ \curvearrowright 㕐, ③ \curvearrowright 㕑, ③ \curvearrowright 㕒, ③ \curvearrowright 㕓, ③ \curvearrowright 㕔, ③ \curvearrowright 㕕, ③ \curvearrowright 㕖, ③ \curvearrowright 㕗, ③ \curvearrowright 㕘, ③ \curvearrowright 㕙, ③ \curvearrowright 㕚, ③ \curvearrowright 㕛, ③ \curvearrowright 㕜, ③ \curvearrowright 㕝, ③ \curvearrowright 㕞, ③ \curvearrowright 㕟, ③ \curvearrowright 㕠, ③ \curvearrowright 㕡, ③ \curvearrowright 㕢, ③ \curvearrowright 㕣, ③ \curvearrowright 㕤, ③ \curvearrowright 㕥, ③ \curvearrowright 㕦, ③ \curvearrowright 㕧, ③ \curvearrowright 㕨, ③ \curvearrowright 㕩, ③ \curvearrowright 㕪, ③ \curvearrowright 㕫, ③ \curvearrowright 㕬, ③ \curvearrowright 㕭, ③ \curvearrowright 㕮, ③ \curvearrowright 㕯, ③ \curvearrowright 㕰, ③ \curvearrowright 㕱, ③ \curvearrowright 㕲, ③ \curvearrowright 㕳, ③ \curvearrowright 㕴, ③ \curvearrowright 㕵, ③ \curvearrowright 㕶, ③ \curvearrowright 㕷, ③ \curvearrowright 㕸, ③ \curvearrowright 㕹, ③ \curvearrowright 㕺, ③ \curvearrowright 㕻, ③ \curvearrowright 㕼, ③ \curvearrowright 㕽, ③ \curvearrowright 㕾, ③ \curvearrowright 㕿, ③ \curvearrowright 㖀, ③ \curvearrowright 㖁, ③ \curvearrowright 㖂, ③ \curvearrowright 㖃, ③ \curvearrowright 㖄, ③ \curvearrowright 㖅, ③ \curvearrowright 㖆, ③ \curvearrowright 㖇, ③ \curvearrowright 㖈, ③ \curvearrowright 㖉, ③ \curvearrowright 㖊, ③ \curvearrowright 㖋, ③ \curvearrowright 㖌, ③ \curvearrowright

D. Knowledge Comprehensiveness

Theorem 7: Given a decision table $T = (U, C, D, V, f)$, denote $RED(\beta)$ as its reduct set with a given β . The set of attribute combinations obtained by traversing $G(C)$ is denoted as RED^* . $(POS_B^\beta(D) = POS_C^\beta(D)) \wedge ((\forall a \in B)(POS_{(B-\{a\})}^\beta(D) < POS_B^\beta(D)))$ for $\forall B \in RED^*$ and $(POS_B^\beta(D) \neq POS_C^\beta(D)) \vee ((\exists b \in B) \wedge (POS_{(B-\{b\})}^\beta(D) = POS_B^\beta(D) = POS_C^\beta(D)))$ for $\forall B \notin RED^*$, which is the necessary and sufficient condition for $RED(\beta) = RED^*$.

Proof 7: The number of nodes is $2^n - 1$ for a complete graph G of depth n . This implies that there are $2^n - 1$ attribute combinations in $P^{G(C)}$. Then $RED(\beta) \subseteq P^{G(C)}$ in $G(C)$.

By Definition 4, if $RED(\beta) = RED^*$, it follows that

$$\begin{aligned} (RED(\beta) = RED^*) \\ \Rightarrow ((\forall B \in RED^*)(POS_B^\beta(D) = POS_C^\beta(D)) \\ \wedge ((\forall a \in B)(POS_{(B-\{a\})}^\beta(D) < POS_B^\beta(D))) \\ \wedge ((\forall B \notin RED^*)(POS_B^\beta(D) \neq POS_C^\beta(D)) \\ \vee ((\exists b \in B) \wedge (POS_{(B-\{b\})}^\beta(D) = POS_B^\beta(D) \\ = POS_C^\beta(D)))). \end{aligned}$$

For $\forall B \in RED^*$, $(POS_B^\beta(D) = POS_C^\beta(D)) \wedge ((\forall a \in B)(POS_{(B-\{a\})}^\beta(D) < POS_B^\beta(D))$, $B \in RED(\beta)$. For $\forall B \notin RED^*$, $(POS_B^\beta(D) \neq POS_C^\beta(D)) \vee ((\exists b \in B) \wedge (POS_{(B-\{b\})}^\beta(D) = POS_B^\beta(D) = POS_C^\beta(D)))$, $B \notin RED(\beta)$. So $RED(\beta) = RED^*$.

Hence, the theorem follows.

Let $RED(\beta)$ represent the set of reducts, and τ is a mapping from the condition attributes set to the decision attributes set. Then the knowledge from the reduct set can be defined as follow.

$$\Psi(\beta) = \{\tau_B | B \in RED(\beta)\} \quad (8)$$

where each of the elements in $\Psi(\beta)$ is a rule, all of which form a comprehensive knowledge system $K = \{U_D', A, I\}$, where U_D' is U_D without the ω component, and I represents the relation between U_D' and A . Defining $x \in U_D'$, $a \in A$, the notation xIa is introduced to signify that the object x has the attribute a [47].

In an information system, the condition attribute $\forall c \in C$, the condition vector V_C and its size $|V_C|$ [48] are as defined in Equ. (9).

$$\begin{aligned} V_C &= V_{c_1} \times V_{c_2} \times \cdots \times V_{c_{|C|}} \\ |V_C| &= \prod_{i=1}^{|C|} |V_{c_i}| \end{aligned} \quad (9)$$

where $|V_C|$ is the number of objects with different values that can theoretically exist in the condition vector, while $|C(U)|$ is the actual number of objects with different values that exists for a given information system. Since the attribute values can be formulated as vectors of Cartesian products of attribute values in the information system, they can be regarded as an extension of the information system [20], [21].

Definition 8: [Information system comprehensiveness] Information system comprehensiveness is the ratio of the actual

and theoretical number of objects with different values, as defined in Equ. (10).

$$\xi = \frac{|C(U)|}{|V_C|} \quad (10)$$

if $\xi = 1$, the system is called a comprehensive information system.

Definition 9: [Knowledge system comprehensiveness] In a knowledge system, the number of objects with different value is $|\Psi(\beta)|$ in the condition vector, and hence knowledge system comprehensiveness, $\hat{\xi}$, can be expressed as:

$$\hat{\xi} = \frac{|\Psi(\beta)|}{|V_C|} \quad (11)$$

if $\hat{\xi} = 1$, the system is called a comprehensive knowledge system.

Definition 10: [Basic rule ratio] Some of the rules in $|\Psi(\beta)|$ already exist in $|C(U)|$. These rules are called basic rules and explicit. The ratio of knowledge extraction, referred to as the basic rule ratio, is defined as

$$\hat{\xi} = \frac{|C(U) \cap \Psi(\beta)|}{|C(U)|} \quad (12)$$

If $\hat{\xi} = 1$ in a comprehensive knowledge system, then the system contains all basic rules from the given information system.

E. Knowledge Structure

We define a structure for a comprehensive knowledge system to illustrate a complete reduct set, the corresponding knowledge and their relationships.

Definition 11: [Reduct chain] All possible attribute subsets for $2^{|C|}$ form a complete lattice. The subset $\{c_1, \dots, c_{|C|}\}$ is the maximum node of the complete lattice, while the empty set Φ is the minimum node. $\forall B \subset C$ and B is a reduct, the chain which passes through B is called a reduct chain from the minimum to the maximum node in the lattice.

The reducts in RED form a complete reduct set lattice. A Hasse diagram of the attributes for $\{c_1, c_2, c_3, c_4\}$ is shown in Figure 2(a) for a given β . These are twenty-four attribute chains, for example $\Phi - c_1 - c_1, c_3 - c_1, c_2, c_3 - c_1, c_2, c_3, c_4$. If $\{c_1, c_3, c_4\}$ and $\{c_2, c_4\}$ are considered as reducts, the two chains highlighted in red are the reduct chains. A Hasse diagram of a complete reduct set is shown in Figure 2(b).

In a comprehensive knowledge system, a formal context with complete reducts can be used to reveal the internal relationships between knowledge from multiple perspectives. The knowledge lattice is obtained from the formal context [47]. Its minimum node is an empty set, and the maximum node is a complete attributes set with corresponding attribute values. For $RED(\beta) = \{B_1, \dots, B_r\}$, the lower the value of $|\bigcup_{i=1}^r B_i| - |\bigcap_{i=1}^r B_i|$, the less the dispersion of complete reducts.

Definition 12: [Double-layer Complete Structure] A double-layer complete structure consists of two layers, in which one is a complete reduct set lattice and the other is a comprehensive knowledge lattice. Set reduct $B \in RED$ and $B^d = B \cup \{d\}$.

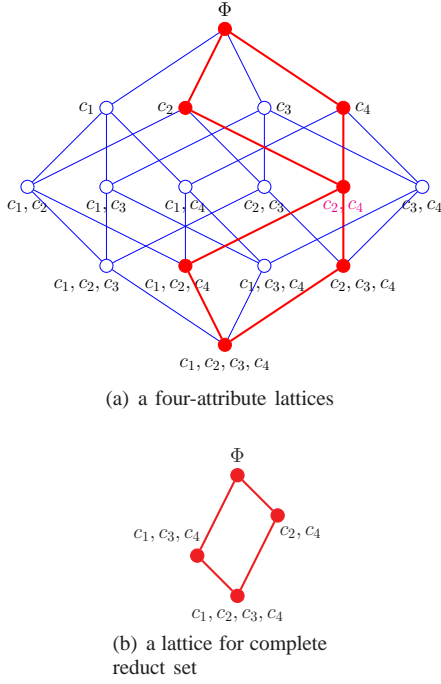


Fig. 2: A complete reduct lattice

In a comprehensive knowledge lattice, each node has a lattice connotation, in which c_i^j denotes that the value of attribute c_i is j , similarly d^j denotes that the value of the decision attribute d is j . Set lattice connotation B' is from an upper-neighbor $\iota_{B'}$ of the maximum node in a comprehensive knowledge lattice, where B' is B^d with corresponding attribute values and denote their relationship as $B' \in^* B^d$. The set of nodes on the chains passing node $\iota_{B'}$, excluding the minimum and maximum nodes, forms a set $\Upsilon(\iota_{B'})$ in a comprehensive knowledge lattice. The relationship between the complete reduct set and the comprehensive knowledge lattices is denoted by “ \rightarrow^* ”. If $B' \in^* B^d$, $B \rightarrow^* \Upsilon(\iota_{B'})$.

Property 1: Some double-layer structure properties are summarized as follows.

- 1) $B'_1 \cap B'_2 \neq \emptyset \Rightarrow B_1 \cap B_2 \neq \emptyset$;
- 2) $B'_1 \cap B'_2 = \emptyset \Leftrightarrow B_1 \cap B_2 = \emptyset$.
- 3) $\Upsilon(\iota_{B'_1}) \cap \Upsilon(\iota_{B'_2}) \neq \emptyset \Rightarrow B'_1 \cap B'_2 \neq \emptyset \Rightarrow B_1 \cap B_2 \neq \emptyset$.

where $B_1, B_2 \in RED$ and $B'_1 \in^* B_1^d, B'_2 \in^* B_2^d$.

IV. KNOWLEDGE EXTRACTION METHODOLOGY

In this section, our complete variable precision reduction (CVPR) algorithm is presented in detail. The computational time complexity and completeness of the methodology are also analyzed and proven.

A. Distribution Table Abstracting and Attribute Rank Dynamic Updating

Given a decision table T , we transform it into β -decision table T_β , in which the decision value of x_i according to a decision monotonicity criterion is adjusted by Equ. (4). Then only object x_i in $\forall [x_i]_c$ is collected and

$\omega(x_i)$ is computed by Equ. (5) in T_β . We get the distribution table $T_D = (U_D(\omega), C, D, V_\beta, f_\beta)$, where $U_D = \{(x_1, \omega(x_1)), \dots, (x_i, \omega(x_i)), \dots, (x_m, \omega(x_m))\}$. It is our distribution table abstracting strategy, which helps us to reduce the considered universe (objects) significantly compared to the original decision table through the equivalence class. From Theorem 2 it follows that this strategy can keep the equivalence between obtaining reducts from T_D and T . An abstracting proportion denotes the percentage reduction of objects from T to T_D as follows.

$$\varsigma = \frac{|U| - |U_D(\omega)|}{|U|} \quad (13)$$

where $|U|$ and $|U_D(\omega)|$ are the numbers of objects in T and T_D , respectively.

Consider here an attribute set H , which is \emptyset initially. We combine incrementally an attribute c into H from the attribute set $C - H$ (i.e. $c \in (C - H)$) in distribution table $T_D = (U_D(\omega), C, D, V_\beta, f_\beta)$ to obtain a reduct. The combination priority of any attribute ($c \in C - H$) will be mainly dependent on its attribute rank. If $\exists [x_i]_H$ with $|[x_i]_H/\omega| \neq 1$, we denote the objects with $\omega(x_i) = 1$ in $[x_i]_H$ with $|[x_i]_H/\omega| \neq 1$ as S_{obj} . In the next loop, if $|[x_i]_H/\omega| = 1$, but $\exists [x_i]_H$ with $\omega(x_i) = 1 \wedge |[x_i]_H/D| \neq 1$, we denote the objects with the same decision values in all $[x_i]_H$ with $\omega(x_i) = 1 \wedge |[x_i]_H/D| \neq 1$ as S_{obj} . If there are multiple different decision values in these equivalence classes, other objects with other different decision values would be denoted as S_{obj} one by one in the next loops in our algorithm. At worst, it will loop $|D| + 1$ times, while at best, it will only loop once, as will be discussed in Subsection IV-B. If S_{obj} has been denoted, our binary classifying label φ is determined as follows.

$$f(x, \varphi) = \begin{cases} 1 & \text{if } x \in S_{obj}, \\ 0 & \text{otherwise.} \end{cases} \quad (14)$$

We use c to partition $[x_i]_H$ with $|[x_i]_H/\varphi| \neq 1$, and then obtain $[x_i]_c'$. Then the attribute rank $r(c)$ can be calculated as

$$r(c) = - \left| \bigcup_{|[x_i]_H/\varphi| \neq 1 \wedge |[x_i]_c'/\varphi| \neq 1 \wedge f(x_i, \varphi) = 1} \{x_i\} \right| \quad (15)$$

It means that $r(c)$ is the negative of the number of φ -inconsistency equivalence classes with respect to attribute c against $[x_i]_H$. The larger $r(c)$ is, the higher the combination priority of attribute c . When $r(c)$ is the same for two attributes priority is determined according to the from-left-to-right order in T_D . If $r(c) = 0$, $H \cup c$ is a candidate reduct, since all objects have been partitioned clearly. If all the equivalence classes with respect to $H \cup c$ satisfy Theorem 2, then this candidate reduct is a reduct according to Definition 2.

Let U_D , $|n'|$ and $|m'|$ represent the universe of discourse, its number of objects and the number of condition attributes in the distribution table, respectively. The pseudo-code of our attribute rank updating (ARU) algorithm for genealogical brotherhood is illustrated in Algorithm 1.

We analyze the time complexity of ARU algorithm as follows. Because the loop from Steps 1 to 9 occurs at most $|C|$ times, the time complexity is $O(|U||C| + \sum_{i=1}^{|C|} (M_i - m_i + 1))$.

ALGORITHM 1: Attribute Rank Updating (ARU) algorithm

input : $[x_i]_H$ with φ
output: attribute rank of a genealogical brotherhood

- 1 **for** $i = 1$ **to** $|m'|$ **do**
- 2 Obtain the maximum and minimum values of c_i
 $(\forall c_i \in C - H)$ for all objects in $[x_i]_H$, denote them as M_i and m_i ;
- 3 Create $M_i - m_i + 1$ queues;
- 4 **for** $j = 1$ **to** $|n'|$ **do**
- 5 Assign $f(x_j, c_i) - m_i$ into the corresponding queue;
- 6 **end**
- 7 Set $r(c_i) = 1$
- 8 Calculate $r(c_i)$ by Equ. (15);
- 9 **end**
- 10 Sort the attributes by $r(c_i)$ and output.

Usually, $\max_{1 \leq i \leq |C|} (M_i - m_i + 1) < |U| \Rightarrow (|U||C| + \sum_{i=1}^{|C|} (M_i - m_i + 1)) \leq (|U||C| + |U||C|)$. Therefore the time complexity of the loop is $O(|U||C|)$, and hence the time complexity of Algorithm 1 is $O(|U||C|)$.

B. Hierarchical Binary Classifying and Genealogical Tree Pruning

The distribution state of objects is determined once the information system is given. Attribute reduction is essential to find some subsets of attributes (i.e. reducts) to partition the objects distinctly through updating the equivalence classes with respect to different tentative attribute combinations. According to Theorem 2, the partition can be implemented by multi-binary classifying of the equivalence classes with $||[x_i]_B/D| \neq 1 \wedge \omega = 1$ hierarchically.

Using the genealogical binary tree of attributes to partition the objects distinctly, the hierarchical binary classifying (HBC) algorithm illustrated in Algorithm 2 is proposed. Here, Ω represents the condition, i.e. the attribute combination Z containing c and its younger siblings satisfies $||[x_i]_Z/\varphi| = 1$ for $\forall [x_i]_Z$, and Y represents the corresponding attributes, i.e. the attributes in the path from the N_f node of the current node to its ancestor.

From Theorem 6, Lemma 1 immediately follows.

Lemma 1: $\forall Z' \subseteq Z$, if $||[x_i]_Z/\varphi| \neq 1$, $||[x_i]_{Z'}/\varphi| \neq 1$.

Lemma 1 provides us with a pruning opportunity. At Step 7 in Algorithm 2, our pruning strategy is triggered in G if $\exists Z \Rightarrow ||[x_i]_Z/\varphi| = 2$, since the attributes at these pruned nodes cannot be included in any reduct. In the same way, our pruning strategy is also triggered at Step 6 if rank $r = 0$. This is very helpful in terms of reducing computational complexity, and enables multiple reducts to be determined with significantly reduced computational effort.

Note that the output of the HBC algorithm is the candidate multiple reduct set RED' for the binary classifying. The validity of each of the candidate reducts is checked using Theorem 2 in our complete attribute reduction algorithm, as

ALGORITHM 2: Hierarchical Binary Classifying (HBC) algorithm

input : H and $[x_i]_H$ with φ
output: Candidate multiple reducts set RED'

- 1 Call Algorithm 1 to calculate rank r of every attribute in $C - H$;
- 2 Generate the ancestor branches according to their attribute ranks;
- 3 **do**
- 4 Visit an ancestor following the ancestor branch one-by-one;
- 5 Obtain the attribute combination set Z ;
- 6 **if** $r \neq 0$ **then**
- 7 **if** Ω is satisfied **then**
- 8 Current attributes are collected into set H_A ;
- 9 Obtain the attribute combination set Y ;
- 10 Current objects U_O are partitioned into $[x_j]_Y$ in $[x]_Y$ which are $||[x_j]_Y/\varphi| = 2$;
- 11 Calculate rank r of every attribute in H_A with U_O by calling Algorithm 1;
- 12 Generate the left child node according to its attribute ranks;
- 13 **else**
- 14 Goto the youngest sibling in the brotherhood;
- 15 **end**
- 16 **end**
- 17 **else**
- 18 Collect the attributes in this genealogical generation path as a candidate reduct and add it into RED' ;
- 19 **end**
- 20 **end**
- 21 **while** all ancestor nodes are visited;

will be discussed in detail in Subsection IV-C. Here we further analyze the time complexity of Algorithm 2 as follows. To execute Algorithm 1, Steps 1 and 11 take no longer than $|U||C|$ time. The useful attributes and objects obtained using Steps 8 and 10 is dominated by $O(|U||C|)$. Since the best, worst and average case time complexity for pre-rank of the binary tree is $O(1)$, $O(2^{|C|})$ and $O(2^{|C|}/|C|)$, respectively, and the average case time complexity through Steps 3 to 21 is $O(2^{|C|}|U|)$, it can be concluded that the average case time complexity of Algorithm 2 is $O(2^{|C|}|U|)$.

C. Knowledge Extraction

In order to obtain a complete reduct set, we have to perform four tasks as follows. Firstly, we transform the considered decision table of the information system into its distribution table. Secondly, attribute rank is calculated dynamically when any attribute combination set is obtained for the reducts. Thirdly, a genealogical binary tree is generated dependent on the attribute rank. Fourthly, the objects in the positive region of the information system will be binary classified hierarchically. During binary classifying, the attributes not included in the

current considered attribute combination will be combined into the current considered attribute combination following any branch of the genealogical tree from the root to the attribute node with rank $r = 0$. After completing the four tasks, all candidate reducts are checked and redundancies removed, and the algorithm outputs the complete reduct sets. Our variable precision reduction (CVPR) algorithm is summarized in Algorithm 3.

ALGORITHM 3: Complete Variable Precision Reduction (CVPR) Algorithm

```

input :  $T$  and  $\beta$ 
output: Complete reduct set  $RED$ 
1 Formulate the distribution table  $T_D$ ;
2 Candidate complete reduct set  $RED' = \emptyset$  and complete
  reduct set  $RED = \emptyset$ ;
3 do
4   Choose the first candidate reduct as  $H$  in  $RED'$ ;
5   Current objects are partitioned into all of  $[x_i]_H$  in
      $[x]_H$ ;
6   Assign  $\varphi$  values of the objects using Equ. (14);
7   Call Algorithm 2 to obtain the candidate reduct set
      $RED'_H$  set;
8   if  $RED'_H \neq null$  then
9     Set  $RED'_H = \{\hat{R}_1, \dots, \hat{R}_j, \dots, \hat{R}_n\}$ ;
10    for  $j = 1 \rightarrow n$  do
11      if all of  $[x_i]_{\hat{R}_j}$  satisfy Theorem 2 then
12        Union  $\hat{R}_j \cup H$  into  $RED$ ;
13      else
14        Union  $\hat{R}_i \cup H$  into the end of  $RED'$ 
15      end
16    end
17  end
18 end
19 Remove the first candidate reduct from  $RED'$ ;
20 while  $RED' \neq null$ ;
21 Remove the redundant reduct(s) in  $RED$ ;

```

In Algorithm 3, Step 1 takes $O(|U||C|)$ time to formulate T_D . The algorithm chooses the attribute sets at least once, on average $2^{|C|}/|C|$ times and at most $2^{|C|}$ times. Therefore the operation has $O(2^{2|C|}|U|/|C|)$ average case time complexity from Steps 3 to 20. Step 21 involves removing reducts with redundant attributes which can be done in $O(4^{|C|})$ time. Therefore the best, worst and average case time complexity of Algorithm 3 is $O(|U||C|)$, $O(4^{|C|}|U||C|)$ and $O(4^{|C|}|U|/|C|)$, respectively.

In general, $|C| < |U|$, however, the rate of exponential growth with respect to C is far faster than the quadratic growth with U , hence in Algorithm 3 the number of attributes has more influence than the number of objects in terms of computational complexity. The pruning strategy is a very important mechanism for reducing the number of objects and consequently decreasing algorithm time complexity.

Theorem 8: $RED(\beta)$ extracted by the CVPR algorithm is the complete reduct set.

Proof 8: From Theorem 7, the complete reduct set is a subset of $P^{G(C)}$ in the genealogical binary tree $G(C)$. According to Lemma 1, the genealogical paths of these attributes in $G(C)$ are pruned during hierarchical binary classifying from Steps 7 to 16 in Algorithm 2 once there is no candidate reduct in the paths. The redundant attribute combinations are removed at Step 21 in Algorithm 3. The theorem follows.

The reduct lattice and its comprehensive knowledge system can be built once the complete reduct set $RED(\beta)$ is obtained by the CVPR algorithm. The comprehensive knowledge system $K = \{U_D', A, I\}$ is considered as a formal context. Then c_i^j and d^i in every knowledge lattice are determined through formal concept analysis [47]. The double-layer structure of the comprehensive knowledge system is established as in Definition 12.

V. EXPERIMENTS, RESULTS AND DISCUSSION.

In this section, our complete attribute reduction algorithm is evaluated based on an implementation in the C language and a computational environment consisting of an Intel® Core™ i5-3230M CPU @2.60GHz processor with 4G memory. We begin by providing a relatively simple example to illustrate comprehensive knowledge extraction and then demonstrate the performance of our algorithm on 30 benchmark datasets with different instances and attributes.

A. Benchmark Problems

To illustrate the effectiveness and performance of the proposed CVPR algorithm, we evaluate its performance on 30 well-known benchmark datasets from the UC Irvine machine learning repository¹. Some of these datasets (e.g. heart, iris, glass) are frequently used to test classification methods. We also consider some of the newer datasets, such as Connectionist Bench, Fertility, and MAGIC Gamma Telescope. The average number of attributes in the datasets is 12 and the maximum number is 60. The average number of objects is 1,980, with ten datasets having more than 1,000 objects and the maximum number is 19,020. The results obtained with different β values are presented and discussed in Subsection V-B. In this section we consider the datasets listed in Table I with $\beta = 1$ to enable a fair comparison with the alternative approaches considered.

In the case of the heart dataset, the ancestor of its attribute genealogical tree is $c_5 - c_8 - c_{10} - c_4 - c_1 - c_2 - c_3 - c_6 - c_7 - c_9 - c_{11} - c_{12} - c_{13}$. The time performance curve during extraction of complete reducts without pruning is illustrated in Figure 3(a). Searching the left subtree of c_5 , we obtain the reducts from the first to the 53rd reduct. Accordingly, c_8 corresponds to the 54th to the 83rd reduct, c_{10} corresponds to the 84th to the 95th reduct, c_4 corresponds to the 96th to the 106th reduct, and c_1 corresponds to the 107th to the 109th reduct. With regard to the child generation, for example, $c_{10} - c_1$ corresponds to the 84th to the 91st reduct, and $c_{10} - c_4$ corresponds to the 92nd to the 95th reduct.

Figure 3(b) shows the time performance curve comparison during the extraction of complete reducts from the heart

¹<http://archive.ics.uci.edu/ml/>

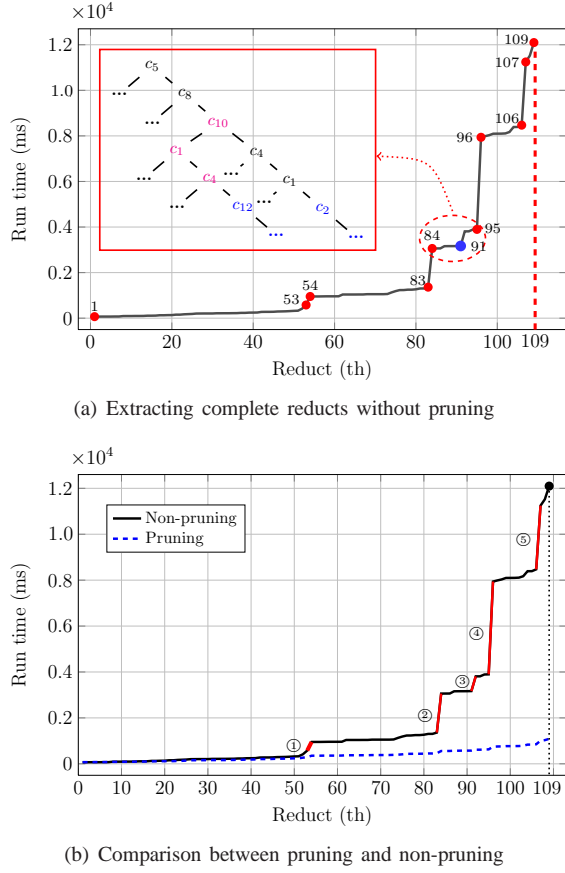


Fig. 3: Reduction performance curves from the heart dataset

dataset with and without pruning. The pruning strategy saves significantly on computational time without omitting any reduct. When traversing the ancestors $c_5 - c_8 - c_{10} - c_4 - c_1 - c_2 - c_3 - c_6 - c_7 - c_9 - c_{11} - c_{12} - c_{13}$, the attribute nodes from c_2 to c_{13} do not satisfy the Ω condition, hence all of them are pruned. Even traversing the first 5 attribute nodes (i.e., $c_5 - c_8 - c_{10} - c_4 - c_1$), the complete traverse procedure takes much longer on five slopes, denoted ①, ②, ③, ④ and ⑤, respectively. For example, to obtain the 95th reduct after traversing c_{10} , takes nearly 3000 ms to get the next reduct (i.e. the 96th reduct) without pruning (the child branches of c_{10}). Since c_{10} 's child node c_4 has eight younger siblings, there are $2^8 - 1$ nodes to be traversed and calculated. Even when sharing one father node, child branch pruning also makes a difference. For example, under the ancestor c_{10} , it takes nearly 1000 ms from traversing its child c_1 to traversing its child c_4 without pruning, as shown in Figures 3(a) and 3(b). In general, there is a positive correlation between the time cost and the number of nodes. The idea of the pruning strategy is that if one attribute node cannot be included in a reduct, then none of its child nodes can be included in the reduct either.

There are three kinds of attribute node in Figure 4, the black ones which are traversed normally, the red one which are skipped, and the blue ones which are pruned. Since the red nodes are rank $r(c) = 0$, it is impossible for the candidate reduct to include more attributes. The blue ones do not satisfy

the Ω condition, and therefore are not valid to include in any reduct. In other words, the red and blue nodes will not generate child nodes, i.e. our pruning strategy is triggered. For example, traversing the path $c_5 \frown c_4 \frown c_9$, we get c_9 with rank $r(c) = 0$, the candidate reduct $\{c_5, c_4, c_9\}$ is obtained, and the left child nodes of c_9 are not transversed. Instead c_{10} is transversed. Note that the attributes of child nodes of one node are the same as its younger siblings, only with different ranks. Since the child node attribute combination set c_2, c_3, c_{12}, c_{13} of c_6 in the attribute set $\{c_5, c_4, c_6\}$ cannot partition its objects, it is not possible for its subset to partition the objects innerrably, hence the branch $c_2 - c_3 - c_{12} - c_{13}$ is pruned.

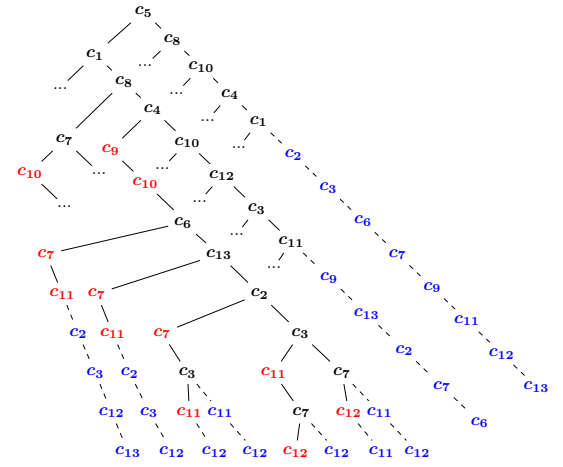


Fig. 4: Part of the heart dataset genealogical tree

In Figure 4, there are two paths $c_7 - c_{11} - c_2 - c_3 - c_{12} - c_{13}$ and $c_6 - c_{13} - c_2 - c_3 - c_7 - c_{11} - c_{12}$. Although the attributes are the same with the exception of c_6 , they have different attribute ranks. They provide dynamic opportunities to join the candidate reduct(s), which is obviously different from the importance of attributes in the model [49]. This is a static quantity and only equivalent to our attribute ranks in the ancestor branch in Figure 4. In the heart dataset there is a reduct $\{c_5, c_8, c_7, c_{10}\}$. According to the classical importance of attributes approach to obtaining the reduct, after selecting c_5 and c_8 , $c_{10}, c_4, c_1, c_2, c_3$, and c_6 would be considered successively. While we select the attributes following the genealogical tree generated through these attribute ranks, after selecting c_5 and c_8 , c_7 should be considered, since our attribute ranks are dynamic. After c_5 and c_8 are selected the objects which need to be considered are reduced with the remaining attributes in a new distribution table. The attribute ranks are updated accordingly, which is very helpful for obtaining the reduct(s) successfully.

In Figure 4, there are two different genealogical generation paths $c_5 \frown c_4 \frown c_3 \frown c_7 \frown c_{12}$ and $c_5 \frown c_4 \frown c_7 \frown c_{12}$, with the former contained in the latter. The former is redundant with respect to the latter. The reason is that when combining the attribute set with different sequences to obtain the candidate reduct, they both partition all the objects equally. Therefore it is necessary to remove the redundancies in the collected reduct set.

In Figure 5, the left subfigure shows the comparison be-

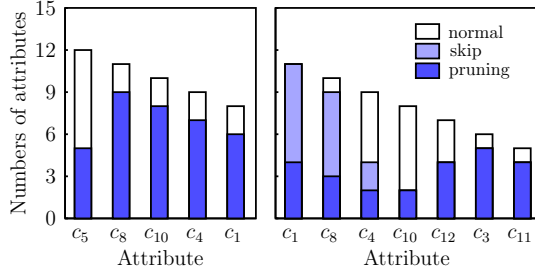


Fig. 5: Node counts for the different node types during pruning

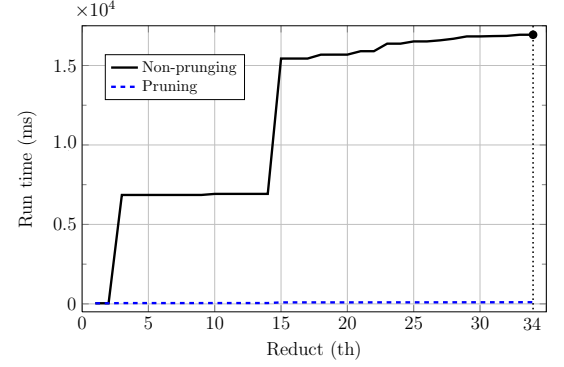
tween the number of normally traversed nodes and the number of pruned nodes about the child nodes of the five first attributes in the ancestor branch for the heart dataset. There are 12 nodes in the next generation of c_5 , as shown in the left bar of the left subfigure, no node is skipped, five pruned, and seven (i.e. $c_1, c_8, c_4, c_9, c_{12}, c_3, c_{11}$) normally traversed. For the next generation of c_5 , the right subfigure of Figure 5 shows further the comparison among the three kinds of node in terms of their visit states. It is clear that the number of normally traversed nodes is a small portion of the nodes in the whole genealogical binary tree in our algorithm, with most nodes either skipped or pruned. Our pruning strategy reduces computational costs significantly. Figure 6 shows more performance comparison curves during reduct extraction for the seismic-bump, Credit Approvals and ILPD datasets.

TABLE I: Considered datasets and the reduct number

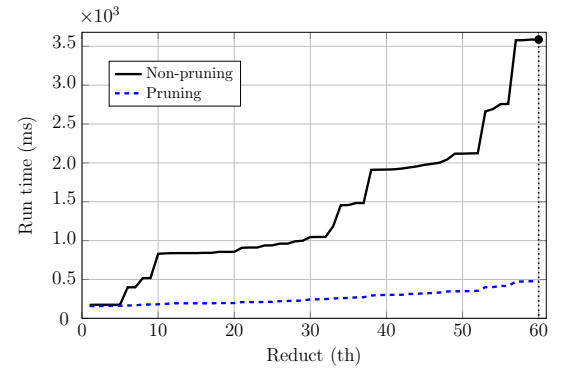
Data set	N^O	N^A	N^R
Credit Approval	690	15	60
heart	270	13	109
Ionosphere	351	34	5759
iris	150	4	4
Lenses	24	4	0
Liver Disorders	345	6	9
Statlog (German Credit Data)	1000	20	846
Glass	214	9	18
Abalone	4177	8	25
Auto MPG	398	8	5
banknote	1372	4	6
Blood Transfusion	748	4	2
Breast Cancer Wisconsin	699	9	19
Breast Tissue	106	9	8
Connectionist Bench	208	60	1314
Contraceptive Method Choice	1473	9	0
Ecoli	336	7	7
EEG Eye	14980	14	597
Fertility	100	9	12
ILPD	583	10	56
Knowledge Modeling Data	403	5	8
MAGIC Gamma Telescope	19020	10	32
Qualitative_Bankruptcy	250	6	8
seeds	210	7	21
seismic-bumps	2584	18	34
SPECT Heart	80	22	725
Thoracic Surgery Data	470	16	14
Wine Quality (red)	1599	11	227
Wine Quality (white)	4898	11	127
Yeast	1484	8	4

^O Number of Objects.^A Number of Condition Attributes.^R Number of Reducts.

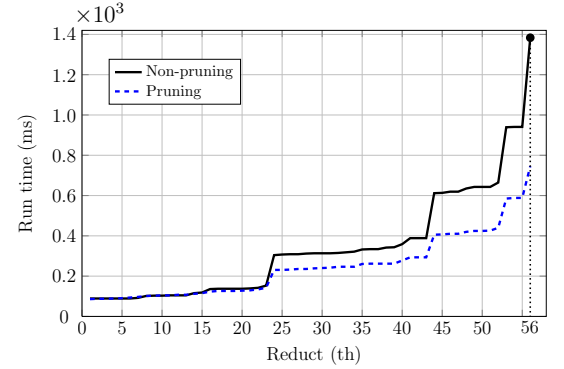
In the seeds dataset, the three decision values are the three



(a) seismic-bumps dataset



(b) Credit Approval dataset



(c) ILPD dataset

Fig. 6: Time performance curves during the extraction of reducts

different varieties of wheat: Kama, Rosa and Canadian. The objects with a decision value of 1 are selected and their classifying value, φ set to 1 at Step 6 in Algorithm 3. After the first calling of Algorithm 2 at Step 7, all 21 reducts are obtained, because $RED' = \emptyset$ at Step 20, which leads directly to Step 21 and then to the end of the whole algorithm. When using the classic classification quality γ [38] to evaluate the attribute's importance (indicating the attribute's impact on all decision classes), there is little difference between all attributes. However our attribute ranks are calculated by considering only two classes during any binary classifying with binary classifying label φ . The significant differences between attributes are very helpful in determining straightforwardly

whether the attribute and its attribute combinations should be included in a candidate reduct or not. For the Contraceptive Method Choice dataset, the search stops once the first binary classifying is completed, since it is impossible to include any attribute to obtain a reduct. In other words, there is no reduct in the dataset. The binary classifying strategy speeds up distinguishing whether there is a candidate reduct or not. Of course, the binary classifying strategy also makes it possible to generate the genealogical binary tree with the attribute ranks. The fourth column in Table I shows the number of complete reducts in each dataset identified by the CVPR algorithm.

In Table I, the highest abstracting proportion defined in Equ. (13) reaches 58.80% for the Qualitative_Bankruptcy dataset. The proportion is more than 30%, for the Blood Transfusion and Breast Cancer Wisconsin datasets, while the proportions for the SPECT Heart, Wine Quality (red) and Wine Quality (white) datasets are 18.75%, 15.01% and 19.13%, respectively.

Table II provides a comparison of the results obtained using the CVPR algorithm and the WADF (worst-attribute-drop-first) algorithm [16]. For the relatively small scale datasets, i.e. credit approval, heart, iris and liver disorders, CVPR and WADF algorithms both determined all of the reducts, while only the CVPR algorithm identified the complete reduct set for the relatively large scale datasets, i.e. Ionosphere and Statlog.

In WADF, the maximum number of multi-reducts that can be identified is the number of non-core attributes in the seed reduct, and the reducts which the algorithm can identify is strongly dependent on the seed reducts. For example, WADF is able to extract all reducts from the Credit Approval and heart datasets, but not from the Ionosphere and Statlog datasets.

TABLE II: Comparisons about the number of obtained multi-reducts

dataset	WADF [16]	ϑ	CVPR	ϑ
Credit Approval	60	100%	60	100%
heart	109	100%	109	100%
iris	4	100%	4	100%
Lenses	0	-	0	-
Liver Disorders	9	100%	9	100%
Ionosphere	32	5.56%	5759	100%
Statlog	16	18.9%	846	100%

To compare the performance of the CVPR and FSRS (fuzzy swarm rough set) algorithms [45], we firstly test a classical dataset, i.e. the Glass dataset, in which there are six decision values. Figure 7 shows the performance of FSRS with respect to identifying multiple reducts. The swarm size was 30, and the maximum number of iterations was 108. The execution time was 213s (the numbers above the vertical dotted lines are the execution times of the corresponding run number in seconds). In total 11 reducts, $(\{c_1, c_2\}, \{c_1, c_4\}, \{c_1, c_5\}, \{c_1, c_6\}, \{c_1, c_7\}, \{c_2, c_4\}, \{c_2, c_5\}, \{c_2, c_7\}, \{c_3, c_5\}, \{c_3, c_7\}, \text{ and } \{c_4, c_7\})$, were obtained, corresponding to run numbers 1, 2, 3, 4, 5, 7, 14, 16, 17, 19 and 20. In the case of the other runs no new reduct were obtained. The reduct completeness ratio ϑ is finally stopped at 61.1% using FSRS. In fact, there are 18 reducts in this dataset, hence 7

reducts, $(\{c_5, c_7, c_9\}, \{c_4, c_5, c_6\}, \{c_4, c_5, c_8\}, \{c_4, c_5, c_9\}, \{c_2, c_3, c_6, c_8\}, \text{ and } \{c_2, c_3, c_6, c_9\})$, were not identified. The main reason that FSRS failed to identify all reducts is that it is a stochastic search algorithm and hence cannot guarantee finding the complete set of reducts in a finite number run times.

Figure 8 shows the time performance curve when using the CVPR algorithm to extract all 18 reducts from the Glass dataset. Here all 18 reducts are obtained within 593 ms reflecting the fact that CVPR algorithm is a deterministic algorithm that considers distribution information and employs effective pruning strategies to reduce the computational overhead.

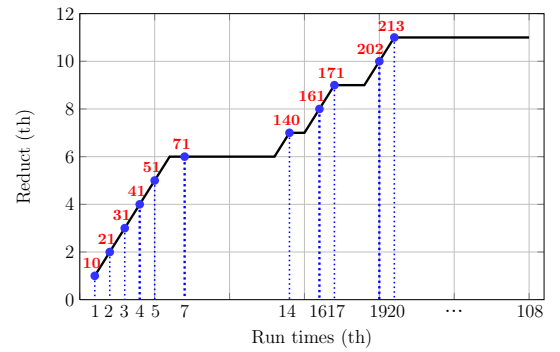


Fig. 7: Multiple reducts processing using FSRS

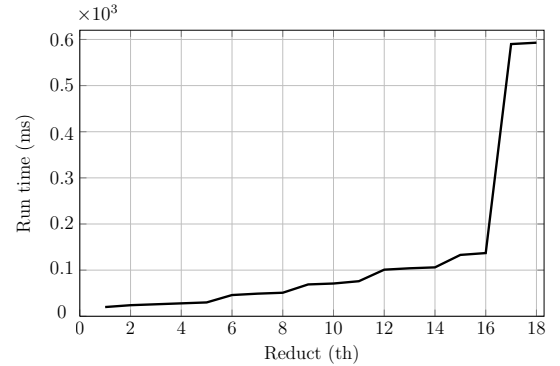


Fig. 8: Reduction performance curve using CVPR from the Glass dataset

The CVPR algorithm is designed based on the theory of comprehensive knowledge extraction introduced in the paper. It is a deterministic search procedure in which dynamic attribute rank is used as heuristical information to traverse its genealogical tree, such that the completeness of the reduct set is ensured. This contrasts with which is different obviously from WADF and FSRS which do not guarantee completeness. Furthermore, CVPR algorithm is optimised with respect to computational overhead through the use of distribution table abstracting, hierarchical binary classifying and genealogical tree pruning.

The numbers of complete reducts for each of the 30 benchmark datasets is listed in Table I, with all reducts satisfying Definition 2. The number of objects in the datasets has little influence on the algorithm search time and completeness of reducts. The reason is that our distribution table collects only one object from each equivalence class in a given information

system, and then only the objects in the subsequent equivalence classes with $|\{x_i\}_H/D| \neq 1 \wedge \omega = 1$ are considered during binary classification. The number of attributes in the datasets has a relatively larger influence on CVPR algorithm performance. The pruning strategy is very helpful in obtaining sound and complete reducts. The average case time complexity of CVPR is $O(4^{|C|}|U|/|C|)$, which also verifies further the theoretical analysis of the time complexity of algorithms and theoretical proofs of completeness in Section IV.

B. From Information Systems to Double-layer Lattice Structures

The previous section illustrated the effectiveness and performance of the proposed CVPR algorithm. Here we provide an illustrative example of how a given information system can be represented as double-layer lattice structures.

Table III shows an instance of a decision table in which the condition attributes are c_1, c_2, c_3, c_4 and c_5 , and d is the decision attribute. Its condition equivalence classes are $\{x_1\}$, $\{x_2, x_6, x_8, x_{10}\}$, $\{x_3\}$, $\{x_4, x_9, x_{11}\}$, $\{x_5\}$ and $\{x_7\}$. The universe of discourse is $\{x_1, x_2, x_3, x_4, x_5, x_7\}$ in T_D .

The $POS_C^\beta(D)$ and complete reduct sets corresponding to different values of β are shown in Table IV. When $\beta = 0.70$, x_6 is added to $POS_C^\beta(D)$. If β is changed to 0.60, x_9 is also within $POS_C^\beta(D)$. Because $POS_C^\beta(D)$ does not change between $\beta = 1$ and $\beta = 0.80$, the complete reduct set is also unchanged. For the complete reduct set with $\beta = 0.7$, the comprehensive knowledge system is illustrated in Table V.

TABLE III: Decision table

objects	c_1	c_2	c_3	c_4	c_5	d
x_1	1	1	1	2	1	1
x_2	1	1	1	0	1	1
x_3	0	1	0	1	0	1
x_4	0	1	1	2	1	1
x_5	1	1	1	0	0	2
x_6	1	1	1	0	1	1
x_7	0	0	0	1	1	2
x_8	1	1	1	0	1	1
x_9	0	1	1	2	1	1
x_{10}	1	1	1	0	1	2
x_{11}	0	1	1	2	1	2

TABLE IV: Complete reduct sets for Table III

β	$POS_C^\beta(D)$	Reduct set
1.00	$\{x_1, x_3, x_5, x_7\}$	$\{\{c_1, c_4, c_5\}\}$
0.80	$\{x_1, x_3, x_5, x_7\}$	$\{\{c_1, c_4, c_5\}\}$
0.70	$\{x_1, x_2, x_3, x_5, x_7\}$	$\{\{c_1, c_2, c_5\}, \{c_1, c_4, c_5\}, \{c_1, c_3, c_5\}\}$
0.60	$\{x_1, x_2, x_3, x_4, x_5, x_7\}$	$\{\{c_4, c_5\}, \{c_1, c_2, c_5\}, \{c_3, c_5\}\}$

When β is set to three different values, three comprehensive knowledge systems are extracted accordingly. Their completeness metric results with different β are illustrated in Table VI. The multi-knowledge system ξ is the same, because it is determined by the original information system, while the knowledge system comprehensiveness $\hat{\xi}$ is influenced by its

TABLE V: Comprehensive knowledge system with $\beta = 0.7$

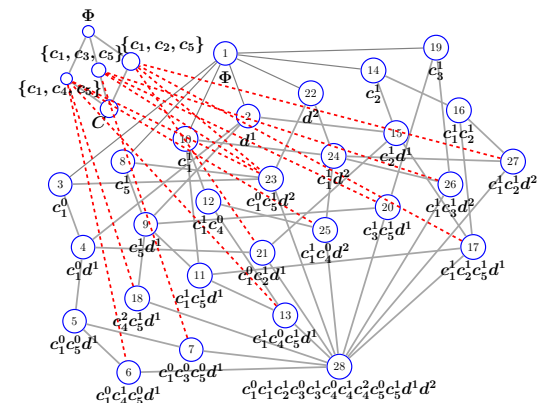
No.	c_1	c_2	c_3	c_4	c_5	d
x_1	1	1	*	*	1	1
x_2	1	1	*	*	1	1
x_3	0	1	*	*	0	1
x_4	0	1	*	*	1	1
x_5	1	1	*	*	0	2
x_7	0	0	*	*	1	2
x_{10}	1	1	*	*	1	2
x_{11}	0	1	*	*	1	2
<hr/>						
x_1	1	*	*	2	1	1
x_2	1	*	*	0	1	1
x_3	0	*	*	1	0	1
x_4	0	*	*	2	1	1
x_5	1	*	*	0	0	2
x_7	0	*	*	1	1	2
x_{10}	1	*	*	0	1	2
x_{11}	0	*	*	2	1	2
<hr/>						
x_1	1	*	1	*	1	1
x_2	1	*	1	*	1	1
x_3	0	*	0	*	0	1
x_4	0	*	1	*	1	1
x_5	1	*	1	*	0	2
x_7	0	*	0	*	1	2
x_{10}	1	*	1	*	1	2
x_{11}	0	*	1	*	1	2

TABLE VI: Completeness metric results with β

β	ξ	$\hat{\xi}$	$\hat{\xi}$
1.00	0.125	0.500	1.000
0.70	0.125	0.896	1.000
0.60	0.125	1.000	1.000

precision parameter β . In the classical rough set model (i.e. $\beta = 1$), ξ is only 0.5, while $\hat{\xi} = 1$ when β decreases to 0.6. Everywhere, the basic rule ratio $\hat{\xi}$ is 1, which indicates our algorithm can extract the comprehensive knowledge from the given information system.

The double-layer structures of the comprehensive knowledge system with $\beta = 0.7$ values are presented in Figure 9. The complete reduct set lattice is depicted in the upper left while the comprehensive knowledge lattice is shown in

Fig. 9: Double-layer structure of comprehensive knowledge system for $\beta = 0.7$

the lower right of each plot in Figure 9. The relationship \in^* between the two types of lattice is highlighted using dashed red lines.

$RED = \{\{c_1, c_2, c_5\}, \{c_1, c_4, c_5\}, \{c_1, c_3, c_5\}\}$ when $\beta = 0.7$ in Figure 9. Node ③ is an upper-neighbor of the maximum node 28 and its connotation is c_1^1, c_4^0, c_5^1, d^1 . $\{c_1^1, c_4^0, c_5^1, d^1\} \in^* \{c_1, c_4, c_5, d\}$, so $\{c_1, c_4, c_5\} \rightarrow^* \Upsilon(\textcircled{3}) = \{2, 8, 9, 10, 11, 12, 13\}$. And $\{c_1, c_2, c_5\} \rightarrow^* \Upsilon(\textcircled{7}) = \{2, 8, 9, 10, 11, 14, 15, 16\}$. The connotation of node ⑦ is c_1^1, c_2^1, c_5^1, d^1 . $(\{c_1^1, c_4^0, c_5^1, d^1\} \cap \{c_1^1, c_2^1, c_5^1, d^1\}) = \{c_1^1, c_5^1, d^1\} \Rightarrow (\{c_1, c_4, c_5\} \cap \{c_1, c_2, c_5\} = \{c_1, c_5\})$. $(\Upsilon(\textcircled{3}) \cap \Upsilon(\textcircled{7}) = \{2, 8, 9, 10, 11\}) \Rightarrow (\{c_1, c_4, c_5\} \cap \{c_1, c_2, c_5\} = \{c_1, c_5\})$.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, a theory of complete variable precision knowledge extraction has been developed. The concept of complete reduct set and knowledge have been proposed. An efficient complete variable precision reduction (CVPR) algorithm which is able to extract comprehensive knowledge has also been developed. This is achieved by transforming the information system into a distribution table, which only collects one object from each equivalence class in a given information system. It is an essential abstract from the original universe of the object discourse. Then the objects in the distribution table are binary classified hierarchically. A genealogical tree is generated according to the rank of attributes, which facilitates dynamically combining attributes and triggering of a pruning strategy to obtain the complete reduct set with significantly low computational cost. The theoretical proof and experimental results presented illustrate that the CVPR algorithm is complete. The comprehensive knowledge system structure is presented through concept lattices and is visualized using Hasse diagrams.

Our CVPR algorithm is suitable for the variable precision reduction and can also be degenerated to a classic rough set model. The underlying approach can also degenerate to give a single minimum reduct, since the first reduct obtained tends to be the minimum reduct. The earlier a reduct is obtained, the fewer attributes are involved, due to the hierarchical binary classifying strategy employed.

The genealogical tree employed is a left-child-right-sibling binary tree. Binary trees and forests can be converted to each other [50]. Therefore in a genealogical binary tree, the offspring of the nodes in a brotherhood will not be mutually interfering when there are similar ways to search for reducts. It is feasible to design a parallel complete reduction algorithm with mapreduce, which will be considered in future work. It is also of interest to explore the application of CVPR algorithm to the analysis of multiple pathways in brain neural networks and big data networks.

ACKNOWLEDGMENT

The authors sincerely thank the editors and the anonymous reviewers for the very helpful and kind comments with regard to improving the presentation of our paper.

REFERENCES

- [1] X. Wu, H. Chen, G. Wu, J. Liu, Q. Zheng, X. He, A. Zhou, Z.-Q. Zhao, B. Wei, Y. Li, Q. Zhang, and S. Zhang, "Knowledge engineering with big data," *IEEE Intel. Sys.*, vol. 30, no. 5, pp. 46–55, 2015.
- [2] Z. Yu, X. Zhu, H.-S. Wong, J. You, J. Zhang, and G. Han, "Distribution-based cluster structure selection," *IEEE Trans. Cy.*, pp. 1–14, 2016.
- [3] K. P. Sankar and B. C. Debarati, "Granular flow graph, adaptive rule generation and tracking," *IEEE Trans. Cy.*, pp. 1–12, 2016.
- [4] B. Xue, M. Zhang, and W. N. Browne, "Particle swarm optimization for feature selection in classification: A multi-objective approach," *IEEE Trans. Cy.*, vol. 43, no. 6, pp. 1656–1671, Dec. 2013.
- [5] P. P. Kundu and S. Mitra, "Feature selection through message passing," *IEEE Trans. Cy.*, pp. 1–11, 2016.
- [6] P. Maji and P. Garai, "Fuzzy-rough simultaneous attribute selection and feature extraction algorithm," *IEEE Trans. Cy.*, vol. 43, no. 4, pp. 1166–1177, Aug. 2013.
- [7] —, "IT2 fuzzy-rough sets and max relevance-max significance criterion for attribute selection," *IEEE Trans. Cy.*, vol. 45, no. 8, pp. 1657–1668, Aug. 2015.
- [8] H. Chen, T. Li, D. Ruan, J. Lin, and C. Hu, "A rough-set-based incremental approach for updating approximations under dynamic maintenance environments," *IEEE Trans. Knowl. Data En.*, vol. 25, no. 2, pp. 274–284, Feb. 2013.
- [9] S. Patra, P. Modi, and L. Bruzzone, "Hyperspectral band selection based on rough set," *IEEE Trans. Geosci. Remote.*, vol. 53, no. 10, pp. 5495–5503, Oct. 2015.
- [10] Y. Qian, S. Li, J. Liang, Z. Shi, and F. Wang, "Pessimistic rough set based decisions: A multigranulation fusion strategy," *Inf. Sci.*, vol. 264, pp. 196–210, 2014.
- [11] X. Zhu, W. Pedrycz, and Z. Li, "Granular data description: designing ellipsoidal information granules," *IEEE Trans. Cy.*, vol. 46, pp. 1–10, 2016.
- [12] Y. Qian, J. Liang, D. Li, F. Wang, and N. Ma, "Approximation reduction in inconsistent incomplete decision tables," *Knowl.-Based Syst.*, vol. 23, no. 5, pp. 427–433, 2010.
- [13] Z. Pawlak, "Rough sets," *Int. J. Comp. Inf. Sci.*, vol. 11, no. 1-2, pp. 341–356, 1982.
- [14] J. N. Liu, Y. Hu, and Y. He, "A set covering based approach to find the reduct of variable precision rough set," *Inf. Sci.*, vol. 275, pp. 83–100, 2014.
- [15] J. Meng, J. Zhang, and Y. Luan, "Gene selection integrated with biological knowledge for plant stress response using neighborhood system and rough set theory," *IEEE ACM Trans. Comput. Bi.*, vol. 12, no. 2, pp. 433–444, Mar.-Apr. 2015.
- [16] Q. Wu, D. Bell, and M. McGinnity, "Multiknowledge for decision making," *Knowl. Inf. Syst.*, vol. 7, no. 2, pp. 246–266, 2005.
- [17] Q. Wu and D. Bell, "Multi-knowledge extraction and application," in *Proc. 9th Int. Conf. Rough Set., Fuzzy Set., Data Min., and Granul. Comput.*, vol. 2639. Springer, 2003, pp. 574–575.
- [18] K. Kaneiwa, "A rough set approach to multiple dataset analysis," *Appl. Soft Comput.*, vol. 11, no. 2, pp. 2538–2547, 2011.
- [19] A. Skowron and C. Rauszer, "The discernibility matrices and functions in information systems," in *Intell. Decis. Supp.* Springer, 1992, vol. 11 of the series Theory and Decision Library, pp. 331–362.
- [20] K. Pancerz, "Some issues on extensions of information and dynamic information systems," in *Found. Computat. Intel.* Springer, 2009, vol. 15, pp. 79–106.
- [21] Z. Suraj, K. Pancerz, and G. Owsiany, "On consistent and partially consistent extensions of information systems," in *Int. Workshop on Rough Set. Fuzzy Set. Data Min. Granul. Soft Comput.* Springer, 2005, pp. 224–233.
- [22] H. M. Abu-Donia, "Multi knowledge based rough approximations and applications," *Knowl. Based Syst.*, vol. 26, pp. 20–29, 2012.
- [23] M. Gams, "New measurements highlight the importance of redundant knowledge," in *Proc. 4th Int. Conf. Eur. Work. Session Learn.*, 1989, pp. 71–80.
- [24] Y. Ma, "Knowledge reduction algorithm based binary discernibility matrix," *J. Anshan Norm. Univ.*, vol. 5, no. 2, pp. 78–82, 2003.
- [25] Y. Qian, J. Liang, and C. Dang, "Incomplete multigranulation rough set," *IEEE Trans. Man. Cy. A*, vol. 40, no. 2, pp. 420–431, Mar 2010.
- [26] J. Yao and N. Azam, "Web-based medical decision support systems for three-way medical decision making with game-theoretic rough sets," *IEEE Trans. Fuzzy Syst.*, vol. 23, no. 1, pp. 3–15, Feb 2015.
- [27] B. Liu, W. Hsu, and Y. Ma, "Mining association rules with multiple minimum supports," in *Proc. 5th ACM/SIGKDD inter. conf. Knowl. disc. data min.* ACM, 1999, pp. 337–341.

- [28] J. Li, Y. Ren, C. Mei, Y. Qian, and X. Yang, "A comparative study of multigranulation rough sets and concept lattices via rule acquisition," *Knowl.-Based Syst.*, 2015.
- [29] B. Ganter, G. Stumme, and R. Wille, *Formal Concept Analysis: Foundations and Applications*. Berlin, GER: Springer Science & Business Media, 2005.
- [30] P. Jonas, I. Dmitry, O. Sergei, and D. Guido, "Formal concept analysis in knowledge processing: A survey on applications," *Expert Syst. Appl.*, vol. 40, no. 16, pp. 6538–6560, 2013.
- [31] Z. Pawlak and A. Skowron, "Rudiments of rough sets," *Inf. Sci.*, vol. 177, no. 1, pp. 3–27, 2007.
- [32] J. Mi, W. Wu, and W. Zhang, "Approaches to knowledge reduction based on variable precision rough set model," *Inf. Sci.*, vol. 159, no. 3, pp. 255–272, 2004.
- [33] W. Ziarko, "Variable precision rough set model," *J. Comput. Syst. Sci.*, vol. 46, no. 1, pp. 39–59, 1993.
- [34] Y. Yao and X. Deng, "Quantitative rough sets based on subethood measures," *Inf. Sci.*, vol. 267, pp. 306–322, 2014.
- [35] P. P. San, S. H. Ling, Nuryani, and H. Nguyen, "Evolvable rough-block-based neural network and its biomedical application to hypoglycemia detection system," *IEEE Trans. Cy.*, vol. 44, no. 8, pp. 1338–1349, Aug. 2014.
- [36] X. Zhang and D. Miao, "Quantitative information architecture, granular computing and rough set models in the double-quantitative approximation space of precision and grade," *Inf. Sci.*, vol. 268, pp. 147–168, 2014.
- [37] I. Park and G. S. Choi, "A variable-precision information-entropy rough set approach for job searching," *Inf. Sci.*, vol. 48, pp. 279–288, 2015.
- [38] J. Wang and J. Zhou, "Research of reduct features in the variable precision rough set model," *Neurocomputing*, vol. 72, no. 10, pp. 2643–2648, 2009.
- [39] D. Slezak, "Normalized decision functions and measures for inconsistent decision tables analysis," *Fund. Inform.*, vol. 44, no. 3, pp. 291–319, 2000.
- [40] M. Li, C. Shang, S. Feng, and J. Fan, "Quick attribute reduction in inconsistent decision tables," *Inf. Sci.*, vol. 254, pp. 155–180, 2014.
- [41] Y. Yang, D. Chen, and Z. Dong, "Novel algorithms of attribute reduction with variable precision rough set model," *Neurocomputing*, vol. 139, pp. 336–344, 2014.
- [42] Q. Wu, "Multiknowledge and computing network model for decision making and localisation of robots," Ph.D. dissertation, University of Ulster, UK, 2005.
- [43] H. Liu, A. Abraham, and M. Clerc, "Chaotic dynamic characteristics in swarm intelligence," *Appl. Soft Comput.*, vol. 7, no. 3, pp. 1019–1026, 2007.
- [44] Y. Gong, J. Li, Y. Zhou, Y. Li, H. S. Chung, Y. Shi, and J. Zhang, "Genetic learning particle swarm optimization," *IEEE Trans. Cy.*, vol. 46, pp. 1–14, 2016.
- [45] H. Liu, C. Yang, M. Zhang, S. McLoone, and Y. Sun, "A computational intelligence approach to multi-factor analysis of violent crime information systems," *Enterp. Inf. Syst.*, pp. 1–24, 2014.
- [46] Z. Pawlak, "Rough sets and intelligent data analysis," *Inf. Sci.*, vol. 147, no. 1–4, pp. 1–12, 2002.
- [47] B. Ganter and R. Wille, *Formal Concept Analysis: Mathematical Foundations*. Berlin, GER: Springer Science & Business Media, 2012.
- [48] Q. Wu, D. A. Bell, G. Prasad, and T. M. McGinnity, "A distribution-index-based discretizer for decision-making with symbolic AI approaches," *IEEE Trans. Knowl. Data En.*, vol. 19, no. 1, pp. 17–28, Jan 2007.
- [49] P. Maji, "A rough hypercuboid approach for feature selection in approximation spaces," *IEEE Trans. Knowl. Data En.*, vol. 26, no. 1, pp. 16–29, Jan 2014.
- [50] E. Horowitz and S. Sahni, *Fundamentals of Data Structures*. London, UK: Pitman, 1983.

VII. SUPPLEMENTAL MATERIALS I

Table VII is a decision table where $U/C = \{\{x_1, x_3\}, \{x_2\}, \{x_4, x_5, x_7\}, \{x_6, x_8\}\}$. The distribution tables with $\beta = 1$ and $\beta = 0.6$ are shown in Table VIII according to Definition 3. Note that x_4 is in the positive region with $\beta = 0.6$ but it is not with $\beta = 1$. The decision values of x_4 are adjusted by Equ. (4) as shown in Table VIII and the values of ω are obtained according to Equ. (5).

TABLE VII: A decision table

Object	c_1	c_2	c_3	c_4	d
x_1	1	1	1	1	0
x_2	2	3	2	3	0
x_3	1	1	1	1	0
x_4	3	1	2	1	0
x_5	3	1	2	1	1
x_6	4	3	4	2	1
x_7	3	1	2	1	1
x_8	4	3	4	2	2

TABLE VIII: The distribution tables with different β

Object	c_1	c_2	c_3	c_4	$\beta = 1$		$\beta = 0.6$	
					d	ω	d	ω
x_1	1	1	1	1	0	1	0	1
x_2	2	3	2	3	0	1	0	1
x_4	3	1	2	1	0	0	1	1
x_6	4	3	4	2	1	0	1	0

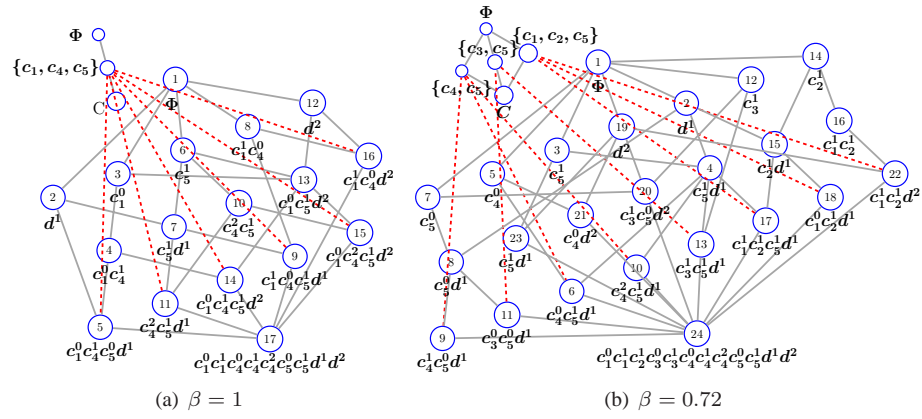
Table IX is a decision table on playing tennis. The condition attributes set is {Outlook, Temp, Humidity, Wind} and {Play} is the decision attribute. *RED* of playing tennis decision table is $\{\{\text{Outlook, Humidity, Wind}\}, \{\text{Outlook, Temp, Wind}\}\}$.

TABLE IX: A decision table about playing tennis

Object	Outlook	Temp	Humid	Wind	Playing
x_1	Sunny	Hot	High	Weak	No
x_2	Sunny	Hot	High	Strong	No
x_3	Overcast	Hot	High	Weak	Yes
x_4	Rain	Mild	High	Weak	Yes
x_5	Rain	Cool	Normal	Weak	Yes
x_6	Rain	Cool	Normal	Strong	No
x_7	Overcast	Cool	Normal	Strong	Yes
x_8	Sunny	Mild	High	Weak	No
x_9	Sunny	Cool	Normal	Weak	Yes
x_{10}	Rain	Mild	Normal	Weak	Yes
x_{11}	Sunny	Mild	Normal	Strong	Yes
x_{12}	Overcast	Mild	High	Strong	Yes
x_{13}	Overcast	Hot	Normal	Weak	Yes
x_{14}	Rain	Mild	High	Strong	No

In Table IX, there are 14 actual objects, i.e. $C(U)=14$. $|v_{\text{Outlook}}| = 3$ because $v_{\text{Outlook}} = \{\text{Sunny, Overcast, Rain}\}$. $V_C = 3 \times 3 \times 2 \times 2 = 36$ by Equ. (9), so the theoretical number of objects is 36. The first and eighth objects have the same condition attribute values according to {Outlook, Humid, Wind}, and the fifth and tenth objects are also the same. Because $|v_{\text{Temp}}| = 3$, the number of the knowledge from {Outlook, Humid, Wind} is $(14 - 2) \times 3 = 36$. Likewise, the cardinality of the knowledge from {Outlook, Temp, Wind} is 24. So $|\Psi(\beta)| = 36$ without redundancy. $\xi = \frac{14}{36} = \frac{7}{18}$, $\hat{\xi} = 1$ and $\hat{\xi} = 1$ according to Eqs. (10) to (12).

The double-layer structures of the comprehensive knowledge system with $\beta = 1$ and $\beta = 0.6$ from Table III are presented in Figure 10.

Fig. 10: Double-layer structure of the comprehensive knowledge system with different β

VIII. SUPPLEMENTAL MATERIALS II

An instance procedure is illustrated in Figure 11. There are two decision values, i.e. 0 and 1, which are partitioned by the red curves. The central white region is $POS_C^\beta(D)$ and the peripheral gray region is $NEG_C^\beta(D)$, as shown in Figure 11. Based on the attribute rank obtained using Algorithm 1, its attribute genealogical binary tree G can be generated to combine attribute sets sequentially. Suppose we get an attribute combination set B_1 , which some equivalence classes satisfy $\omega(x_i) = 1$, i.e., $[x_1]_{B_1}, [x_2]_{B_1}, [x_3]_{B_1}$ while all other satisfy $\omega(x_i) = 0$, i.e., $[x_4]_{B_1}, [x_5]_{B_1}, [x_6]_{B_1}$ in Figure 11(b). Note that $[x_1]_{B_1}$ and $[x_2]_{B_1}$ do not satisfy $|[x_i]_{B_1}/D| = 1$. Now we only focus on these two equivalence classes in the next steps. An object set can be collected from the objects in $[x_1]_{B_1}$ and $[x_2]_{B_1}$ in Figure 11(c) with the attributes $C - B_1$. Accordingly for this object set, all attribute ranks in $C - B_1$ are re-calculated and the corresponding subtree of the attribute genealogical binary tree G also continues to generate further to the current attribute nodes. From this attribute subtree, suppose we get an attribute combination set B_2 to partition $[x_1]_{B_1}$ in Figure 11(c) into $[x_{11}]_{B_2}$ and $[x_{12}]_{B_2}$, and to partition $[x_2]_{B_1}$ into $[x_{21}]_{B_2}, [x_{22}]_{B_2}$ and $[x_{23}]_{B_2}$, as shown in Figure 11(d). Thus far, all equivalence classes of $B_1 \cup B_2$ satisfy Theorem 2. It means that $\forall x_i \in [x_i]_{(B_1 \cup B_2)}, \omega(x_i) = 0$ or $\omega(x_i) = 1 \wedge |[x_i]_{(B_1 \cup B_2)}/D| = 1$. Hence, a reduct is obtained as $B_1 \cup B_2$.

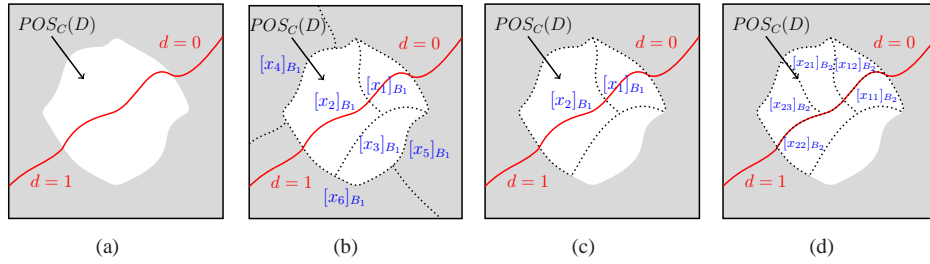


Fig. 11: Multi-binary classification for a reduct

IX. SUPPLEMENTAL MATERIALS III

Considering a simple example to demonstrate our algorithms, Table X and $\beta = 1$ are used as the input of Algorithm 3. $U_D = \{(x_1, 0), (x_2, 1), (x_3, 1), (x_4, 0), (x_6, 1), (x_8, 1)\}$ in T_D can be obtained using Eqs. (4) and (5) in Step 1. The φ values of x_1 and x_4 are assigned 1 by Equ. (14) in Step 6. When calling Algorithm 2 in Step 7, $r(c_4) = 0, r(c_5) = -1, r(c_3) = -1, r(c_2) = -2, r(c_1) = -2$ according to Algorithm 1 using Equ. (15) and $RED'_H = \{\{x_4\}, \{x_5, x_3\}, \{x_5, x_2\}, \{x_3, x_2, x_1\}\}$. Every candidate reduct in RED'_H is checked to see if it is a reduct or not in Step 11. Then $RED = \{\{x_4\}, \{x_5, x_3\}, \{x_5, x_2\}, \{x_3, x_2, x_1\}\}$ and $RED' = \emptyset$ from Steps 12 to 14. The algorithms end and the complete reduct set RED is returned.

TABLE X: A decision table for the algorithm demonstration

Object	c_1	c_2	c_3	c_4	c_5	d
x_1	1	1	1	1	1	1
x_2	1	0	1	0	0	0
x_3	0	1	1	0	0	0
x_4	1	1	2	1	2	1
x_5	1	1	1	1	1	0
x_6	1	1	0	0	0	0
x_7	1	1	2	1	2	0
x_8	1	0	0	0	1	0

X. SUPPLEMENTAL MATERIALS IV

The flowchart of our attribute reduction algorithm is presented in Figure 12.

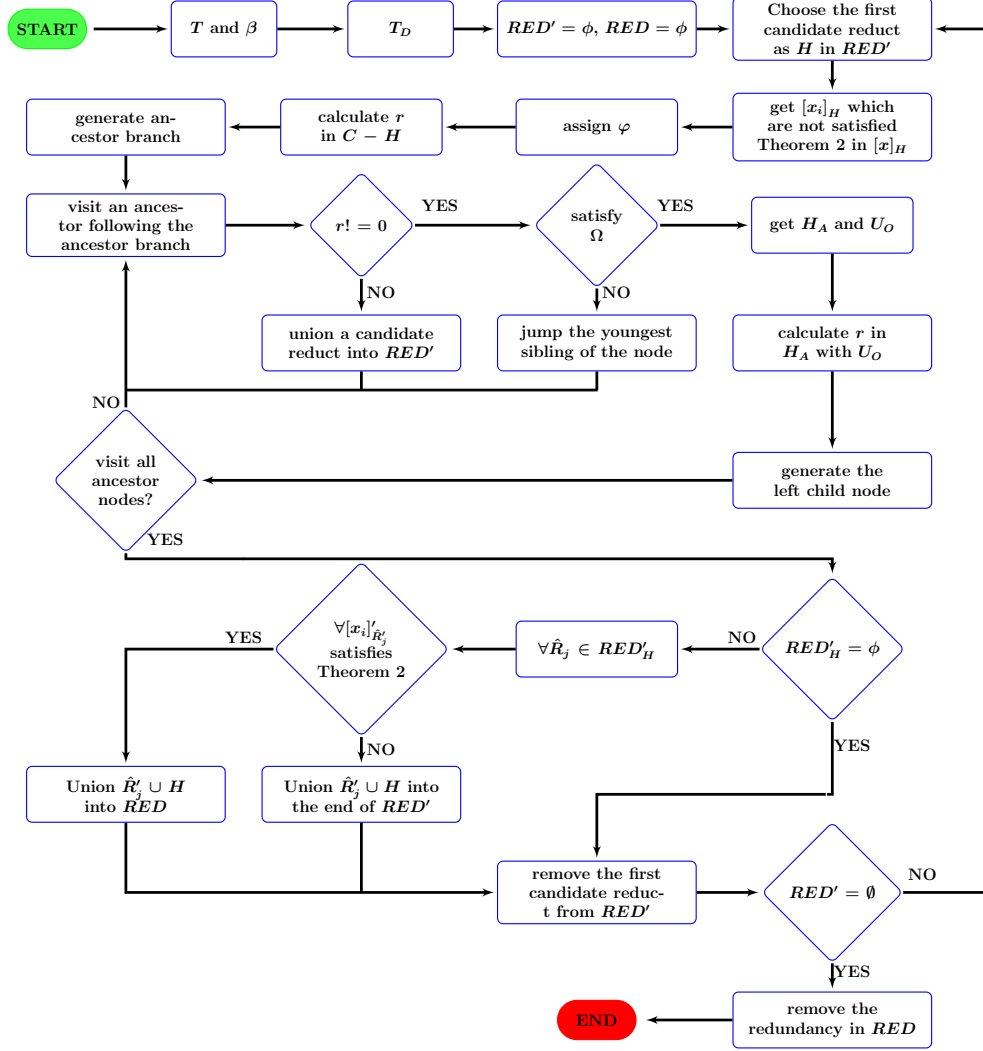


Fig. 12: Flowchart of the complete attribute reduction algorithm