

Article

Efficient Attribute Reduction Algorithms Using Discernibility Attributes for Hierarchical Classification

Ruilin Wei, Nan Zhang *  and Yuxuan He

School of Computer and Control Engineering, Yantai University, Yantai 264005, China; wrl@s.ytu.edu.cn (R.W.); hyx@s.ytu.edu.cn (Y.H.)

* Correspondence: zhangnan@ytu.edu.cn; Tel.: +86-186-1535-1131

Abstract

Attribute reduction, also referred to as feature selection, focuses on seeking a minimal attribute subset and is an important topic in rough set theory (RST). Classical rough set-based attribute reduction is restricted to decision systems with a single label level and fails to obtain attribute reducts across hierarchical label levels, leading to low computational efficiency. To tackle this issue, this paper proposes a novel method to derive bidirectional reducts across label levels. We first establish the reduction relationship from the coarse label level to the fine label level by analyzing the relationships among decision classes at different label levels. Correspondingly, we construct the reduction relationship from the fine label level to the coarse label level by investigating the connections between positive regions across different label levels. Based on these relationships, two efficient attribute reduction algorithms are developed, which can rapidly compute a reduct at one label level from the reduct at another level. Experimental results on twelve UCI datasets demonstrate that the proposed algorithms require less running time than the other four methods while still achieving high classification accuracy.

Keywords: rough sets; attribute reduction; granular computing; feature selection; uncertainty analysis

1. Introduction

The rapid growth in data volume and dimensionality has brought significant challenges for data analysis. Feature selection has become an essential technique for reducing redundancy and improving computational efficiency. In RST [1], feature selection [2] is referred to as attribute reduction [3], which aims to identify a minimal subset of condition attributes while preserving the classification capability of the decision system.

In flat classification structures, various attribute reduction methods have been proposed in RST, such as discernibility matrix-based methods [4,5] and heuristic attribute reduction methods [6,7], etc. Discernibility matrix-based methods have the advantage of being able to obtain all reducts; however, their computational complexity is high. In practical applications, we only need to obtain one reduct of a system rather than all reducts. Based on this consideration, many heuristic attribute reduction algorithms have been developed. These heuristic methods have been designed for different reduction objectives. For example, Zhang et al. [8] proposed an unsupervised feature selection method that integrates information quality, information quantity, and difference degree, enabling feature selection under uncertain information. Chen et al. [9] proposed the granular sequential three-way reduction algorithm to preserve the positive region unchanged. Zhang et al. [10]



Received: 12 March 2026

Revised: 31 March 2026

Accepted: 2 April 2026

Published: 3 April 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

introduced an interactive streaming feature selection method, which provides a new research perspective for feature selection in streaming data. Xu et al. [11] developed an algorithm based on local composite entropy. Ye et al. [12] introduced a weighted temporal model for three-way efficacy evaluation. Zhang et al. [13] proposed a Dempster–Shafer and KNN-based method for multi-source data fusion and feature selection, which provides a new approach for feature selection in complex data. Yuan et al. [14] developed zentropy to characterize the interactions among multiple data levels and designed a corresponding reduction algorithm. In addition to designing new reduction models, many studies have focused on acceleration mechanisms to improve computational efficiency [15,16]. Qian et al. [17] proposed a positive approximation strategy to enhance the computational performance of attribute reduction. Ding et al. [18] developed a parallel incremental method based on attribute trees for large dynamic decision systems. Li et al. [19] introduced a family of quick methods to derive different forms of reducts from inconsistent decision tables. Chen et al. [20] improved reduction efficiency by grouping candidate attributes. Xia et al. [21] utilized local redundancy stability and relative importance to improve both the efficiency and accuracy of reduction. Wang et al. [22] developed a fast algorithm by exploiting inconsistent equivalence classes.

However, these algorithms are mainly designed for flat data and may not be suitable for hierarchical data. Therefore, many researchers have studied hierarchical classification structures in RST. Feng and Miao [23] integrated rough sets with conceptual hierarchical trees to develop a hierarchical rough set model. Wu et al. [24] introduced multi-scale information systems, in which the same attribute may have different values at different scales. In such systems, optimal scale selection [25,26] becomes a key issue for maintaining the effectiveness of attribute reduction. Huang et al. [27] introduced the concept that decision attributes may take different values at different levels and proposed generalized multi-scale decision systems. These studies extend flat data to hierarchical data and provide an essential foundation for attribute reduction in hierarchical classification [28,29].

Based on hierarchical classification structures, many attribute reduction methods have been developed. Zhao et al. [30] proposed a method by exploiting the relationships among sibling nodes. Qiu et al. [31] designed a hierarchical attribute reduction strategy that decomposes the selection process into multiple subtasks according to the hierarchy. Yang et al. [32] introduced stability factors based on discernibility information for reduction. Lin et al. [33] developed a hierarchical attribute reduction method based on inter-category relevance and semantic independence to reduce attribute dependency. In addition, Yang et al. [34] combined hierarchical fuzzy rough sets with three-way decisions to handle uncertainty, and Huang et al. [35] proposed an incremental method for high-dimensional hierarchical data. Apart from reduction methods, several studies focus on the design of hierarchical models and their applications in various domains. Wang et al. [36] presented a hierarchical evaluation method and constructed a multi-granularity decision model. Qian et al. conducted multiple studies on hierarchical classification, including complex classification [37], knowledge discovery [38], decision optimization [39], and privacy protection [40]. Although hierarchical information has been introduced into existing studies, attribute reduction is still performed independently at each label level. Consequently, the reduct obtained at one level cannot be reused at another, resulting in low efficiency.

Unlike most existing attribute reduction methods in hierarchical decision systems, this study does not compute reducts separately at each label level. Instead, it derives a reduct at one label level from a reduct at another label level by computing discernibility attributes and making use of the available information. Specifically, the inclusion relationships of decision classes and positive regions between different label levels are analyzed. Then, discernibility attributes are introduced to distinguish the object pairs that need further

separation. Thus, reduction relationships between different label levels are established. The key contributions of this study are listed below.

- (1) We investigate reduction relationships between different label levels in a hierarchical decision system. By analyzing the relationships among decision classes and positive regions at different label levels, we introduce the concept of discernibility attributes and combine them with the existing reduct at one label level, thereby establishing reduction relationships between the coarse and fine label levels.
- (2) Based on the reduction relationship from the coarse to the fine label level, we propose an algorithm to derive the fine label level reduct from the coarse label level reduct.
- (3) Based on the reduction relationship from the fine to the coarse label level, we propose an algorithm to derive the coarse label level reduct from the fine label level reduct.

Experimental results show that the attribute reduction algorithms proposed in this paper significantly improve time efficiency while maintaining good classification performance.

The remainder of this paper is organized as follows: some concepts are introduced in Section 2. In Section 3, the reduction relationships between different label levels are established. Section 4 presents two efficient attribute reduction algorithms. The corresponding experiments are performed in Section 5. Section 6 is the conclusion.

2. Preliminaries

This section introduces fundamental concepts, including Pawlak rough sets and the hierarchical classification structure.

2.1. Rough Sets

This subsection reviews some concepts of rough set theory used in this paper.

Definition 1 ([1]). Let $DS = (U, C \cup D, V, f)$ denote a decision system, where U is a finite universe of objects, C and D represent the sets of condition attributes and decision attributes, respectively, where $C \cap D = \emptyset$. $V = \bigcup_{c \in C \cup D} V_c$ denotes the attribute value domain, and $f : U \times (C \cup D) \rightarrow V$ is the information function. For simplicity, $\forall s \in U$, $c(s)$ denotes $f(s, c)$.

$\forall Q \subseteq C$, the equivalence relation induced by Q is defined as $IND(Q) = \{(s_u, s_v) \in U \times U \mid \forall c \in Q, c(s_u) = c(s_v)\}$. Accordingly, $U/Q = \{[s_u]_Q \mid s_u \in U\}$, where $[s_u]_Q = \{s_v \in U \mid (s_u, s_v) \in IND(Q)\}$ is the equivalence class containing s_u .

Definition 2 ([1]). Given a DS. $U/D = \{D_1, D_2, \dots, D_{|U/D|}\}$. $Q \subseteq C$ and $D_i \in U/D$, the lower approximation(LA) and the upper approximation(UA) of D_i are defined as follows:

$$\underline{APR}_Q(D_i) = \{s_u \in U \mid [s_u]_Q \subseteq D_i\}, \quad (1)$$

$$\overline{APR}_Q(D_i) = \{s_u \in U \mid [s_u]_Q \cap D_i \neq \emptyset\}. \quad (2)$$

Here, $|U/D|$ represents the total number of equivalence classes generated by D . The positive region of D relative to Q can be defined as

$$POS_Q(D) = \bigcup_{i=1}^{|U/D|} \underline{APR}_Q(D_i). \quad (3)$$

Definition 3 ([41]). Given a DS. $\forall s_u, s_v \in U$, $D = \{d\}$. The discernibility matrix for positive region preservation $DM = (m(s_u, s_v))_{|U| \times |U|}$ is a $|U| \times |U|$ matrix, where the discernibility attribute set $m(s_u, s_v)$ is defined as

$$m(s_u, s_v) = \begin{cases} \{c \in C \mid c(s_u) \neq c(s_v)\}, & \text{condition,} \\ \emptyset, & \text{otherwise.} \end{cases} \quad (4)$$

where $\text{condition} = (s_u \in \text{POS}_C(D) \vee s_v \in \text{POS}_C(D)) \wedge d(s_u) \neq d(s_v)$. This means that at least one of the two objects belongs to the positive region, and their decision values are different.

2.2. Hierarchical Classification Structure

Hierarchical classification is a multi-level decision structure in which decision classes exhibit inclusion relationships across different levels. Wu et al. [24] introduced multi-scale information systems with multiple scales for the attribute values, forming a hierarchical structure from coarse to fine scales. This subsection reviews the hierarchical classification structure and introduces basic definitions.

Definition 4 ([24]). Let $HDS = (U, C \cup \{d^1, d^2, \dots, d^n\}, V, f)$ be a hierarchical decision system, where U is a nonempty finite universe, C is a nonempty finite set of condition attributes, and $D = \{d^1, d^2, \dots, d^n\}$ is a set of decision attributes with $C \cap D = \emptyset$. For the t -th label level, $D^t = \{d^t\}$, where $d^t \in D$ and $t \in \{1, 2, \dots, n\}$.

Each decision attribute $d^t : U \rightarrow V_{d^t}$ maps objects to decision values at label level t . For $1 \leq t_i < t_j \leq n$, there exists a surjective granular information transformation $h^{t_i, t_j} : V_{d^{t_i}} \rightarrow V_{d^{t_j}}$, such that $d^{t_j}(s_u) = h^{t_i, t_j}(d^{t_i}(s_u))$, $\forall s_u \in U$, where $h^{t_i, t_j} = h^{t_j-1, t_j} \circ \dots \circ h^{t_i, t_i+1}$.

In an HDS , if D contains only one decision attribute, the hierarchical decision system degenerates into a classical Pawlak decision system.

Definition 5. Given an HDS , for $t_i < t_j$, $t_i, t_j \in \{1, 2, \dots, n\}$, $D^{t_i} = \{d^{t_i}\}$, $D^{t_j} = \{d^{t_j}\}$, where t_i and t_j correspond to the fine and the coarse label levels, respectively. D^{t_i} is finer than D^{t_j} (or D^{t_j} is coarser than D^{t_i}), denoted by $D^{t_i} \prec D^{t_j}$, if and only if $\forall s_u \in U$, $[s_u]_{D^{t_i}} \subseteq [s_u]_{D^{t_j}}$ and $\exists s_v \in U$, $[s_v]_{D^{t_i}} \subset [s_v]_{D^{t_j}}$.

Definition 6. Given an HDS , $Q \subseteq C$ and $D_i^t \in U/D^t$, the LA and UA of D_i^t are defined as

$$\underline{\text{APR}}_Q(D_i^t) = \{s_u \in U \mid [s_u]_Q \subseteq D_i^t\}, \quad (5)$$

$$\overline{\text{APR}}_Q(D_i^t) = \{s_u \in U \mid [s_u]_Q \cap D_i^t \neq \emptyset\}. \quad (6)$$

The positive region of D^t with respect to Q is defined as

$$\text{POS}_Q(D^t) = \bigcup_{i=1}^{|U/D^t|} \underline{\text{APR}}_Q(D_i^t). \quad (7)$$

Property 1. Given an HDS , for $t_i < t_j$, $t_i, t_j \in \{1, 2, \dots, n\}$, $\forall s_u \in U$, $Q \subseteq C$, the following properties hold:

- (1) $[s_u]_{D^{t_i}} \subseteq [s_u]_{D^{t_j}}$;
- (2) $\text{POS}_Q(D^{t_i}) \subseteq \text{POS}_Q(D^{t_j})$.

In an HDS , different label levels correspond to different decision systems and may require different subsets of condition attributes to preserve decision information. It is

necessary to identify minimal attribute subsets for each label level. The positive region preserving reduct [42] in an *HDS* is defined as follows.

Definition 7. Given an *HDS*, $Q \subseteq C$ and $c \in Q$, Q is called a positive region preserving reduct of the t -th label level in an *HDS* if it satisfies

$$\text{POS}_{Q-\{c\}}(D^t) \neq \text{POS}_Q(D^t), \quad (8)$$

$$\text{POS}_Q(D^t) = \text{POS}_C(D^t). \quad (9)$$

In Definition 7, if Q satisfies only (9), then Q is called a positive region consistent set (PRCS). The reduct of the t -th label level in an *HDS* is defined as $\text{red}(D^t) = Q$. Equivalently, $\text{red}(D^t)$ satisfies $\text{POS}_{\text{red}(D^t)}(D^t) = \text{POS}_C(D^t)$, and each attribute in $\text{red}(D^t)$ is indispensable. For $t_i < t_j$, $t_i, t_j \in \{1, 2, \dots, n\}$, $\text{red}(D^{t_i})$ is defined as a reduct for the fine label level, $\text{red}(D^{t_j})$ is defined as a reduct for the coarse label level.

Example 1. Table 1 presents an *HDS*, $C = \{c_1, c_2, c_3, c_4, c_5\}$, $D = \{d^1, d^2\}$.

While $t = 1$, $\text{POS}_C(D^1) = \{s_2, s_4, s_7\}$. By removing attributes one by one, $\text{POS}_{C-\{c_1\}}(D^1) \neq \text{POS}_C(D^1)$, while $\text{POS}_{C-\{c_i\}}(D^1) = \text{POS}_C(D^1)$ for $i = 2, \dots, 5$. Thus, $\text{red}(D^1) = \{\{c_1, c_3\}, \{c_1, c_4, c_5\}\}$.

While $t = 2$, $\text{POS}_C(D^2) = \{s_1, s_2, s_4, s_6, s_7\}$. Similarly, $\text{POS}_{C-\{c_2\}}(D^2) \neq \text{POS}_C(D^2)$, while $\text{POS}_{C-\{c_i\}}(D^2) = \text{POS}_C(D^2)$ for $i \neq 2$. Thus, $\text{red}(D^2) = \{\{c_2, c_3\}, \{c_2, c_4\}\}$.

Table 1. A hierarchical decision system.

U	c_1	c_2	c_3	c_4	c_5	D	
						d^1	d^2
s_1	2	0	1	1	1	0	0
s_2	1	1	2	2	2	2	1
s_3	2	0	2	2	2	2	1
s_4	2	0	0	1	2	0	0
s_5	2	0	2	2	2	1	0
s_6	2	0	1	1	1	1	0
s_7	1	0	1	1	1	1	0
s_8	2	0	2	2	2	2	1
s_9	2	1	1	1	1	3	1
s_{10}	2	1	1	1	1	4	2

3. The Reduction Relationships Between Different Label Levels in an *HDS*

In classical RST, attribute reduction [43,44] at different label levels in an *HDS* is usually performed independently, without considering the relationships among different label levels. Although this independent computation method can obtain reducts at different label levels, it results in low computational efficiency. In this section, we analyze the relationships among reducts at different label levels and establish that the reduct at one label level can be derived from that at another. This improves the computational efficiency. Section 3.1 investigates the reduction relationship from the coarse to the fine label level, and Section 3.2 investigates the reduction relationship from the fine to the coarse label level. The corresponding two algorithms are presented in Section 4.

3.1. The Reduction Relationship from the Coarse to the Fine Label Level

This subsection analyzes the reduction relationship from the coarse to the fine label level in an *HDS*. Some objects in the positive region of the fine label level can be effectively

distinguished using the reduct obtained from the coarse label level. By analyzing the relationships between decision classes at different label levels, the concept of discernibility attributes is introduced to distinguish those objects in the positive region of the fine label level that cannot be distinguished by the coarse label level reduct. Thus, the reduction relationship from the coarse to the fine label level is established.

Definition 8. Given an HDS, for a label level t , the discernibility matrix for positive region preservation is defined as DM_t , the discernibility attribute set between s_u and s_v in DM_t is defined as $m_t(s_u, s_v)$, where $s_u, s_v \in U$.

The nonempty discernibility attribute sets in DM_{t_i} are analyzed as follows.

By Property 1, for $t_i < t_j$, $t_i, t_j \in \{1, 2, \dots, n\}$, $\forall s_u \in U$, it holds that $[s_u]_{D^{t_i}} \subseteq [s_u]_{D^{t_j}}$. Therefore, the nonempty discernibility attribute set $m_{t_i}(s_u, s_v)$ in DM_{t_i} is discussed in the following cases.

$$(1) \quad \text{If } [s_u]_{D^{t_i}} = [s_u]_{D^{t_j}}, \text{ then} \quad m_{t_i}(s_u, s_v) = m_{t_j}(s_u, s_v). \quad (10)$$

$$(2) \quad \text{If } [s_u]_{D^{t_i}} \subset [s_u]_{D^{t_j}}, \Delta_{t_i, t_j}^D(s_u) = [s_u]_{D^{t_j}} - [s_u]_{D^{t_i}}, \text{ then} \quad (11)$$

$$m_{t_i}(s_u, s_v) = \begin{cases} \{c_k \in C \mid c_k(s_u) \neq c_k(s_v)\}, & (s_u, s_v) \in R_{\text{POS}}^{t_i}, \\ m_{t_j}(s_u, s_v), & \text{otherwise,} \end{cases}$$

$$\text{where } R_{\text{POS}}^{t_i} = \{(s_u, s_v) \mid (s_u \in \text{POS}_C(D^{t_i})) \wedge (s_v \in \Delta_{t_i, t_j}^D(s_u)) \wedge (d^{t_i}(s_u) \neq d^{t_i}(s_v))\}.$$

Based on the analysis of the nonempty discernibility attribute sets in DM_{t_i} , it is found that the discernibility attribute sets for some object pairs are nonempty in DM_{t_i} but empty in DM_{t_j} . It follows that $\forall s_u, s_v \in U$, if $(s_u, s_v) \in R_{\text{POS}}^{t_i}$, then $m_{t_j}(s_u, s_v) = \emptyset$.

Figure 1 illustrates the relationship between objects s_u and s_v from the coarse to the fine label level, where s_u belongs to the positive region at the fine label level, and s_u and s_v are in the same decision class at the coarse label level but in different decision classes at the fine label level.

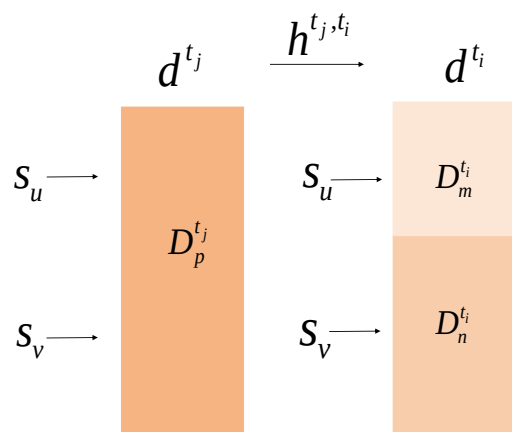


Figure 1. The relationship between objects s_u and s_v from the coarse to the fine label level.

Example 2 illustrates that the discernibility attribute sets for some object pairs are nonempty in DM_{t_i} but empty in DM_{t_j} .

Example 2. In Table 1, the discernibility matrix for positive region preservation is constructed when $t = 1$. By symmetry, it is sufficient to display only the lower triangular part of the discernibility

matrix. The attributes c_1, c_2, c_3, c_4, c_5 are denoted by 1, 2, 3, 4, 5, respectively. The discernibility matrix is given as follows.

In DM_1 , nonempty discernibility attribute sets are highlighted in bold, indicating that they do not appear in DM_2 . $t = 1$, object pairs (s_2, s_9) , (s_4, s_5) , (s_4, s_6) , (s_4, s_7) , and (s_7, s_1) generate nonempty discernibility attribute sets. When the label level is changed from $t = 2$ to $t = 1$, these objects are divided into different decision classes, and the corresponding discernibility attribute sets do not appear in DM_2 .

$$DM_1 = \begin{pmatrix} \emptyset & & & & & & & & \\ \{1, 2, 3, 4, 5\} & \emptyset & & & & & & & \\ \emptyset & \emptyset & \emptyset & & & & & & \\ \emptyset & \{1, 2, 3, 4\} & \{3, 4\} & \emptyset & & & & & \\ \emptyset & \{1, 2\} & \emptyset & \{3, 4\} & \emptyset & & & & \\ \emptyset & \{1, 2, 3, 4, 5\} & \emptyset & \{3, 5\} & \emptyset & \emptyset & & & \\ \{1\} & \{2, 3, 4, 5\} & \{1, 3, 4, 5\} & \{1, 3, 5\} & \emptyset & \emptyset & \emptyset & & \\ \emptyset & \emptyset & \emptyset & \{3, 4\} & \emptyset & \emptyset & \{1, 3, 4, 5\} & \emptyset & \\ \emptyset & \{1, 3, 4, 5\} & \emptyset & \{2, 3, 5\} & \emptyset & \emptyset & \{1, 2\} & \emptyset & \emptyset \\ \emptyset & \{1, 3, 4, 5\} & \emptyset & \{2, 3, 5\} & \emptyset & \emptyset & \{1, 2\} & \emptyset & \emptyset \end{pmatrix}$$

Lemma 1. Given an HDS, $s_u, s_v \in U, s_u \vee s_v \in \text{POS}_C(D^t) \wedge d^t(s_u) \neq d^t(s_v)$. $Q \subseteq C$, Q is a PRCS at the t -th label level if and only if $[s_u]_Q \neq [s_v]_Q$.

Proof. ($\Rightarrow$) Suppose that Q is a PRCS. Then $\text{POS}_Q(D^t) = \text{POS}_C(D^t)$. If $[s_u]_Q = [s_v]_Q$, which implies $d^t(s_u) = d^t(s_v)$. This leads to a contradiction with $d^t(s_u) \neq d^t(s_v)$. Hence, $[s_u]_Q \neq [s_v]_Q$.

($\Leftarrow$) Suppose that Q is not a PRCS. Then $\text{POS}_Q(D^t) \neq \text{POS}_C(D^t)$, $s_u \in \text{POS}_C(D^t) - \text{POS}_Q(D^t)$, $s_v \in [s_u]_Q$. This leads to a contradiction with $[s_u]_Q \neq [s_v]_Q$. Therefore, $\text{POS}_Q(D^t) = \text{POS}_C(D^t)$, Q is a PRCS. $\square$

Based on Lemma 1, the following lemma characterizes the relationship between nonempty discernibility attribute sets and PRCSs.

Lemma 2. Given an HDS, $s_u, s_v \in U, s_u \vee s_v \in \text{POS}_C(D^t) \wedge d^t(s_u) \neq d^t(s_v)$. $Q \subseteq C$, Q is a PRCS at the t -th label level if and only if $Q \cap m_t(s_u, s_v) \neq \emptyset$.

Proof. ($\Rightarrow$) Suppose that Q is a PRCS. By Lemma 1, $s_u, s_v \in U$, $(s_u \in \text{POS}_C(D^t) \vee s_v \in \text{POS}_C(D^t)) \wedge d^t(s_u) \neq d^t(s_v)$, $[s_u]_Q \neq [s_v]_Q$. Hence, there exists $c \in Q$ such that $c(s_u) \neq c(s_v)$, i.e., $Q \cap m_t(s_u, s_v) \neq \emptyset$.

($\Leftarrow$) Suppose that Q is not a PRCS. $s_u \in \text{POS}_C(D^t)$, $s_v \in [s_u]_Q$. which implies $Q \cap m_t(s_u, s_v) = \emptyset$. Hence, Q is a PRCS. $\square$

According to Lemma 2, a PRCS must have a nonempty intersection with every nonempty discernibility attribute set at the fine label level. However, for some object pairs, the discernibility attribute sets are nonempty at the fine label level but empty at the coarse label level. Some of these object pairs cannot be distinguished by the coarse label level reduct. Therefore, when deriving the fine label level reduct from the coarse label level reduct, these object pairs should be considered.

The following definition gives the object pairs that cannot be distinguished with respect to $\text{red}(D^{t_j})$.

Definition 9. Given an HDS, for $t_i < t_j$, $t_i, t_j \in \{1, 2, \dots, n\}$. Then

$$\text{POS}_{diff}^{cf} = \text{POS}_C(D^{t_i}) - \text{POS}_{\text{red}(D^{t_j})}(D^{t_i}). \quad (12)$$

$\forall s_u \in POS_{diff}^{cf}, [s_u]_{D^{t_i}} \subseteq [s_u]_{D^{t_j}}, \Delta_{t_i, t_j}^D(s_u) = [s_u]_{D^{t_j}} - [s_u]_{D^{t_i}}$. Then

$$object^{cf}(s_u) = \Delta_{t_i, t_j}^D(s_u) \cap [s_u]_{red(D^{t_j})}. \quad (13)$$

$\forall s_u \in POS_{diff}^{cf}$ and $s_v \in object^{cf}(s_u)$, (s_u, s_v) is defined as an object pair that cannot be distinguished by $red(D^{t_j})$.

In Definition 9, POS_{diff}^{cf} denotes the difference of the positive region at the fine label level with respect to C and $red(D^{t_j})$. For $\forall s_u \in POS_{diff}^{cf}, \forall s_v \in object^{cf}(s_u)$, (s_u, s_v) is an object pair that cannot be distinguished by $red(D^{t_j})$.

For example, in Table 1, when $red(D^{t_2}) = \{c_2, c_3\}$, $POS_{diff}^{cf} = \{s_7\}$ and $\Delta_{t_1, t_2}^D(s_7) = \{s_1, s_4\}$. The object pair (s_4, s_7) can be distinguished by $red(D^{t_2})$, thus $s_4 \notin object^{cf}(s_7)$, whereas (s_1, s_7) remains indistinguishable by $red(D^{t_2})$. Therefore, $object^{cf}(s_7) = \{s_1\}$.

Based on Definition 9, the pairs of objects that remain indiscernible with respect to $red(D^{t_j})$ are identified. The following definition gives the discernibility attributes for distinguishing these object pairs.

Definition 10. Given an HDS, $\forall s_u \in POS_{diff}^{cf}, s_v \in object^{cf}(s_u)$, $att^{cf}(s_u, s_v) = \{c \in C - red(D^{t_j}) \mid c(s_u) \neq c(s_v)\}$, then the discernibility attributes are defined as

$$att^{cf} = \bigcup_{s_u, s_v} att^{cf}(s_u, s_v). \quad (14)$$

The discernibility attribute set att^{cf} consists of attributes that distinguish object pairs (s_u, s_v) with $s_u \in POS_{diff}^{cf}$ and $s_v \in object^{cf}(s_u)$, which remain indistinguishable for the reduct $red(D^{t_j})$. For each object pair (s_u, s_v) , only one attribute in $att^{cf}(s_u, s_v)$ is required to distinguish them in the reduction process.

Based on discernibility attributes and Lemma 2, the following theorem establishes the reduction relationship from the coarse to the fine label level.

Theorem 1. Given an HDS, for $t_i < t_j$, $t_i, t_j \in \{1, 2, \dots, n\}$, if a coarse label level reduct $red(D^{t_j})$ is given, the fine label level reduct $red(D^{t_i})$ can be derived from it in the following cases.

(1) If $POS_{red(D^{t_j})}(D^{t_i}) = POS_C(D^{t_i})$, then

$$red(D^{t_i}) = red(D^{t_j}) - R, \quad (15)$$

where $R = \{c \in red(D^{t_j}) \mid POS_{red(D^{t_j}) - \{c\}}(D^{t_i}) = POS_C(D^{t_i})\}$.

(2) If $POS_{red(D^{t_j})}(D^{t_i}) \neq POS_C(D^{t_i})$, then

$$red(D^{t_i}) = (red(D^{t_j}) \cup att^{cf}) - R, \quad (16)$$

where $R = \{c \in red(D^{t_j}) \cup att^{cf} \mid POS_{(red(D^{t_j}) \cup att^{cf}) - \{c\}}(D^{t_i}) = POS_C(D^{t_i})\}$.

Proof.

- (1) If $POS_{red(D^{t_j})}(D^{t_i}) = POS_C(D^{t_i})$, then $red(D^{t_j})$ is a PRCS at the t_i -th label level in an *HDS*. To satisfy the conditions of a positive region preserving reduct, $red(D^{t_i})$ is derived from $red(D^{t_j})$ by removing the redundant attribute set R .
- (2) If $POS_{red(D^{t_j})}(D^{t_i}) \neq POS_C(D^{t_i})$, then there exist object pairs (s_u, s_v) with $s_u \in POS_{diff}^{cf}$ and $s_v \in object^{cf}(s_u)$ such that $m_{t_i}(s_u, s_v) \neq \emptyset$ and $red(D^{t_j}) \cap m_{t_i}(s_u, s_v) = \emptyset$. By computation, att^{cf} contains attributes that distinguish these object pairs, so $\forall s_u, s_v \in U$, if $m_{t_i}(s_u, s_v) \neq \emptyset$, then $(red(D^{t_j}) \cup att^{cf}) \cap m_{t_i}(s_u, s_v) \neq \emptyset$. By Lemma 2, $red(D^{t_j}) \cup att^{cf}$ is a PRCS at the t_i -th label level in an *HDS*. To satisfy the conditions of a positive region preserving reduct, $red(D^{t_i})$ is derived from $red(D^{t_j}) \cup att^{cf}$ by removing the redundant attribute set R .

□

Theorem 1 gives the theoretical proof that the reduct at the fine label level can be derived from the reduct at the coarse label level. The reduct at the fine label level does not need to be computed from the beginning. It can be obtained by using the reduct at the coarse label level and adding the needed discernibility attributes. In this method, the reduction relationship from the coarse to the fine label level is established. This result also gives the theoretical basis for the proposed reduction algorithm.

In this subsection, the reduction relationship from the coarse to the fine label level is established. First, the nonempty discernibility attribute sets at the fine label level are analyzed. This analysis shows that for some object pairs, the discernibility attribute sets are nonempty at the fine label level but empty at the coarse label level. This means that these object pairs may not be distinguished by the coarse label level reduct. Then, Lemmas 1 and 2 establish the relationship between nonempty discernibility attribute sets and PRCSs, showing that a PRCS must intersect with nonempty discernibility attribute sets. Accordingly, Definition 9 gives the pairs of objects that remain indiscernible with respect to the coarse label level reduct, and Definition 10 defines the discernibility attributes for distinguishing these object pairs. Finally, based on Lemma 2 and the discernibility attributes, Theorem 1 establishes the reduction relationship from the coarse to the fine label level. The corresponding reduction algorithm from the coarse to the fine label level is presented in Section 4.1.

3.2. The Reduction Relationship from the Fine to the Coarse Label Level

This subsection analyzes the reduction relationship from the fine to the coarse label level in an *HDS*. The fine label level reduct can already distinguish some objects in the positive region of the coarse label level. By analyzing the relationships between positive regions at different label levels, the concept of discernibility attributes is introduced to distinguish those objects in the positive region of the coarse label level that cannot be distinguished by the fine label level reduct. Accordingly, the reduction relationship from the fine to the coarse label level is established.

The nonempty discernibility attribute sets in DM_{t_j} are analyzed as follows.

By Property 1, for $t_i < t_j$, $t_i, t_j \in \{1, 2, \dots, n\}$, it holds that $POS_C(D^{t_i}) \subseteq POS_C(D^{t_j})$. Therefore, the nonempty discernibility attribute set $m_{t_j}(s_u, s_v)$ in DM_{t_j} is discussed in the following cases.

- (1) If $POS_C(D^{t_i}) = POS_C(D^{t_j})$, then

$$m_{t_j}(s_u, s_v) = m_{t_i}(s_u, s_v). \quad (17)$$

(2) If $POS_C(D^{t_i}) \subset POS_C(D^{t_j})$, $\Delta_{t_i, t_j}^{POS} = POS_C(D^{t_j}) - POS_C(D^{t_i})$. Then

$$m_{t_j}(s_u, s_v) = \begin{cases} \{c_k \in C \mid c_k(s_u) \neq c_k(s_v)\}, & (s_u, s_v) \in R_{POS}^{t_j}, \\ m_{t_i}(s_u, s_v), & \text{otherwise,} \end{cases} \quad (18)$$

where $R_{POS}^{t_j} = \{(s_u, s_v) \mid (s_u \in \Delta_{t_i, t_j}^{POS}) \wedge (s_v \notin POS_C(D^{t_i})) \wedge (d^{t_j}(s_u) \neq d^{t_j}(s_v))\}$.

Based on the analysis of the nonempty discernibility attribute sets in DM_{t_j} , it is found that the discernibility attribute sets for some object pairs are nonempty in DM_{t_j} but empty in DM_{t_i} . It follows that $\forall s_u, s_v \in U$, if $(s_u, s_v) \in R_{POS}^{t_j}$, then $m_{t_i}(s_u, s_v) = \emptyset$.

Figure 2 illustrates the relationship between objects s_u and s_v from the fine to the coarse label level, where s_u belongs to the positive region at the coarse label level, s_u and s_v are not in the positive region at the fine label level, and they belong to different decision classes at the coarse label level.

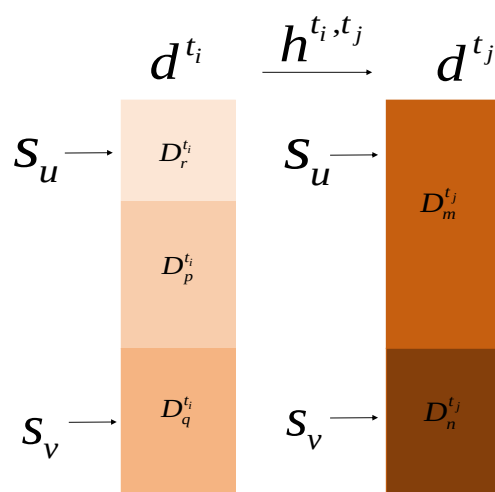


Figure 2. The relationship between objects s_u and s_v from the fine to the coarse label level.

Example 3 illustrates that the discernibility attribute sets for some object pairs are nonempty in DM_{t_j} but empty in DM_{t_i} .

Example 3 (Continuation of Example 1). In Table 1, the discernibility matrix for positive region preservation is constructed when $t = 2$. By symmetry, it is sufficient to display only the lower triangular part of the discernibility matrix. The attributes c_1, c_2, c_3, c_4, c_5 are denoted by 1, 2, 3, 4, 5, respectively. The discernibility matrix is given as follows.

$$DM_2 = \left(\begin{array}{cccccccccccc} \emptyset & & & & & & & & & & & \\ \{1, 2, 3, 4, 5\} & \emptyset & & & & & & & & & & \\ \{3, 4, 5\} & \emptyset & \emptyset & & & & & & & & & \\ \emptyset & \{1, 2, 3, 4\} & \{3, 4\} & \emptyset & & & & & & & & \\ \emptyset & \{1, 2\} & \emptyset & \emptyset & \emptyset & & & & & & & \\ \emptyset & \{1, 2, 3, 4, 5\} & \{3, 4, 5\} & \emptyset & \emptyset & \emptyset & & & & & & \\ \emptyset & \{2, 3, 4, 5\} & \{1, 3, 4, 5\} & \emptyset & \emptyset & \emptyset & \emptyset & & & & & \\ \{3, 4, 5\} & \emptyset & \emptyset & \{3, 4\} & \emptyset & \{3, 4, 5\} & \{1, 3, 4, 5\} & \emptyset & & & & \\ \{2\} & \emptyset & \emptyset & \{2, 3, 5\} & \emptyset & \{2\} & \{1, 2\} & \emptyset & \emptyset & & & \\ \{2\} & \{1, 3, 4, 5\} & \emptyset & \{2, 3, 5\} & \emptyset & \{2\} & \{1, 2\} & \emptyset & \emptyset & \emptyset & & \end{array} \right)$$

In DM_2 , nonempty discernibility attribute sets are highlighted in bold, indicating that they do not appear in DM_1 . $POS_C(D^1) = \{s_2, s_4, s_7\}$ and $POS_C(D^2) = \{s_1, s_2, s_4, s_6, s_7\}$, so $\Delta_{1,2}^{POS} = \{s_1, s_6\}$. That is, an object in $\Delta_{1,2}^{POS}$ is paired with an object not in $POS_C(D^{t_1})$, and

the two objects have different decision values when $t = 2$, which results in additional nonempty discernibility attribute sets in DM_2 .

According to Lemma 2, a PRCS must have a nonempty intersection with nonempty discernibility attribute sets at the coarse label level. However, for some object pairs, the discernibility attribute sets are nonempty at the coarse label level but empty at the fine label level. Some of these object pairs cannot be distinguished by the fine label level reduct. Therefore, when deriving the coarse label level reduct from the fine label level reduct, these object pairs should be considered.

The following definition gives the object pairs that cannot be distinguished with respect to $red(D^{t_i})$.

Definition 11. Given an HDS, for $t_i < t_j$, $t_i, t_j \in \{1, 2, \dots, n\}$. Then

$$POS_{diff}^{fc} = POS_C(D^{t_j}) - POS_{red(D^{t_i})}(D^{t_j}). \quad (19)$$

$\forall s_u \in POS_{diff}^{fc}$, $\Delta_{t_i, t_j}^{D^C - P}(s_u) = [s_u]_{D^{t_j}}^C - POS_C(D^{t_i})$. Then

$$object^{fc}(s_u) = \Delta_{t_i, t_j}^{D^C - P}(s_u) \cap [s_u]_{red(D^{t_i})}. \quad (20)$$

$\forall s_u \in POS_{diff}^{fc}$ and $s_v \in object^{fc}(s_u)$, (s_u, s_v) is defined as an object pair that cannot be distinguished by $red(D^{t_i})$.

In Definition 11, $[s_u]_{D^{t_j}}^C$ is defined as the complement of the equivalence class $[s_u]_{D^{t_j}}$ in U . POS_{diff}^{fc} denotes the difference of the positive region at the coarse label level with respect to C and $red(D^{t_i})$. For $\forall s_u \in POS_{diff}^{fc}$, $\forall s_v \in object^{fc}(s_u)$, (s_u, s_v) is an object pair that cannot be distinguished by $red(D^{t_i})$.

Based on Definition 11, the pairs of objects that remain indiscernible with respect to $red(D^{t_i})$ are identified. The following definition gives the discernibility attributes for distinguishing these object pairs.

Definition 12. Given an HDS, $\forall s_u \in POS_{diff}^{fc}$, $s_v \in object^{fc}(s_u)$, $att^{fc}(s_u, s_v) = \{c \in C - red(D^{t_i}) \mid c(s_u) \neq c(s_v)\}$, then the discernibility attributes are defined as

$$att^{fc} = \bigcup_{s_u, s_v} att^{fc}(s_u, s_v). \quad (21)$$

The discernibility attribute set att^{fc} consists of attributes that distinguish object pairs (s_u, s_v) with $s_u \in POS_{diff}^{fc}$ and $s_v \in object^{fc}(s_u)$, which remain indistinguishable for the reduct $red(D^{t_i})$. For each object pair (s_u, s_v) , only one attribute in $att^{fc}(s_u, s_v)$ is required to distinguish them in the reduction process.

Based on discernibility attributes and Lemma 2, the following theorem establishes the reduction relationship from the fine to the coarse label level.

Theorem 2. Given an HDS, for $t_i < t_j$, $t_i, t_j \in \{1, 2, \dots, n\}$, given a fine label level reduct $red(D^{t_i})$, the coarse label level reduct $red(D^{t_j})$ can be derived from it in the following cases.

(1) If $POS_{red(D^{t_i})}(D^{t_j}) = POS_C(D^{t_j})$, then

$$red(D^{t_j}) = red(D^{t_i}) - R, \quad (22)$$

- where $R = \{c \in \text{red}(D^{t_i}) \mid \text{POS}_{\text{red}(D^{t_i}) - \{c\}}(D^{t_j}) = \text{POS}_C(D^{t_j})\}$.
- (2) If $\text{POS}_{\text{red}(D^{t_i})}(D^{t_j}) \neq \text{POS}_C(D^{t_j})$, then

$$\text{red}(D^{t_j}) = (\text{red}(D^{t_i}) \cup \text{att}^{fc}) - R, \quad (23)$$

where $R = \{c \in \text{red}(D^{t_i}) \cup \text{att}^{fc} \mid \text{POS}_{(\text{red}(D^{t_i}) \cup \text{att}^{fc}) - \{c\}}(D^{t_j}) = \text{POS}_C(D^{t_j})\}$.

Proof.

- (1) If $\text{red}(D^{t_i})$ satisfies $\text{POS}_{\text{red}(D^{t_i})}(D^{t_j}) = \text{POS}_C(D^{t_j})$, then $\text{red}(D^{t_i})$ is a PRCS at the t_j -th label level. To satisfy the conditions of a positive region preserving reduct, $\text{red}(D^{t_j})$ is derived from $\text{red}(D^{t_i})$ by removing the redundant attribute set R .
- (2) If $\text{red}(D^{t_i})$ satisfies $\text{POS}_{\text{red}(D^{t_i})}(D^{t_j}) \subset \text{POS}_C(D^{t_j})$, then there exist object pairs (s_u, s_v) with $s_u \in \text{POS}_{\text{diff}}^{fc}$ and $s_v \in \text{object}^{fc}(s_u)$ such that $m_{t_j}(s_u, s_v) \neq \emptyset$ and $\text{red}(D^{t_i}) \cap m_{t_j}(s_u, s_v) = \emptyset$. By computation, att^{fc} contains attributes that distinguish these object pairs, so $\forall s_u, s_v \in U$, if $m_{t_j}(s_u, s_v) \neq \emptyset$, then $(\text{red}(D^{t_i}) \cup \text{att}^{fc}) \cap m_{t_j}(s_u, s_v) \neq \emptyset$. By Lemma 2, $\text{red}(D^{t_i}) \cup \text{att}^{fc}$ is a PRCS at the t_j -th label level. To satisfy the conditions of a positive region preserving reduct, $\text{red}(D^{t_j})$ is derived from $\text{red}(D^{t_i}) \cup \text{att}^{fc}$ by removing the redundant attribute set R .

□

Theorem 2 gives the theoretical proof that the reduct at the coarse label level can be derived from the reduct at the fine label level. The reduct at the coarse label level does not need to be computed from the beginning. It can be obtained by using the reduct at the fine label level and adding the needed discernibility attributes. In this method, the reduction relationship from the fine to the coarse label level is established. This result also gives the theoretical basis for the proposed reduction algorithm.

In this subsection, the reduction relationship from the fine to the coarse label level is established. First, the nonempty discernibility attribute sets at the coarse label level are analyzed. This analysis shows that for some object pairs, the discernibility attribute sets are nonempty at the coarse label level but empty at the fine label level. This means that these object pairs may not be distinguished by the fine label level reduct. Based on this, Definition 11 gives the pairs of objects that remain indiscernible with respect to the fine label level reduct, and Definition 12 defines the discernibility attributes for distinguishing these object pairs. Finally, based on Lemma 2 and the discernibility attributes, Theorem 2 establishes the reduction relationship from the fine to the coarse label level. The corresponding reduction algorithm from the fine to the coarse label level is presented in Section 4.2.

4. Attribute Reduction Algorithms Between Different Label Levels

Based on Theorems 1 and 2, two efficient attribute reduction algorithms for an *HDS* are proposed. One obtains a reduct from the coarse to the fine label level, and the other obtains a reduct from the fine to the coarse label level.

4.1. An Attribute Reduction Algorithm from the Coarse to the Fine Label Level

Based on the reduction relationship from coarse to fine provided in Theorem 1, Algorithm 1 is proposed, which mainly consists of the following parts. First, the positive region of the fine decision attribute set with respect to the coarse label level reduct and the condition attribute set is computed. Then, the algorithm compares the two positive regions. If they are not equal, their difference is computed. Otherwise, the fine-label level reduct can be directly derived from the coarse-label level reduct by removing redundant

attributes. The algorithm scans each object in this set. For each such object, the set of objects that need to be distinguished from it is computed according to Definition 9. Then, the object is compared with each object in this set, and a discernibility attribute that can distinguish the two objects is selected from the attributes not included in the coarse label level reduct and added to the coarse label level reduct. After that, redundant attributes are removed, and the reduct at the fine label level is returned.

The time complexity of Algorithm 1 is $O(|U| \cdot |C| + |U'| \cdot |U''| \cdot |C'| + |U| \cdot |C''|^2)$, where $U' = POS_{diff}^{cf}$, $U'' = object^{cf}(s_u)$, $C' = C - red(D^{t_j})$, and $C'' = red(D^{t_j})$. Here, C'' denotes the current reduct, which is initialized as $red(D^{t_j})$ and may be updated by adding the selected discernibility attributes.

Algorithm 1: An Attribute Reduction Algorithm from the coarse to the fine label level (LTCF-PR)

Input: A hierarchical decision system $HDS = (U, C \cup \{d^t \mid t = 1, 2, \dots, n\}, V, f)$; a reduct $red(D^{t_j})$.

Output: A reduct $red(D^{t_i})$.

```

1 Compute  $POS_C(D^{t_i}), POS_{red(D^{t_j})}(D^{t_i})$ ;
2 if  $POS_{red(D^{t_j})}(D^{t_i}) \neq POS_C(D^{t_i})$  then
3    $POS_{diff}^{cf} \leftarrow POS_C(D^{t_i}) - POS_{red(D^{t_j})}(D^{t_i})$ ;
4    $att^{cf} \leftarrow \emptyset$ ;
5   foreach  $s_u \in POS_{diff}^{cf}$  do
6     Calculate  $object^{cf}(s_u)$  by Definition 9;
7     foreach  $s_v \in object^{cf}(s_u)$  do
8       foreach  $c \in C - red(D^{t_j})$  do
9         if  $c(s_u) \neq c(s_v)$  then
10           $att^{cf} \leftarrow att^{cf} \cup \{c\}$ ;
11          break;
12    $red(D^{t_i}) \leftarrow red(D^{t_j}) \cup att^{cf}$ ;
13 foreach  $c \in red(D^{t_j})$  do
14   if  $POS_{red(D^{t_j})-\{c\}}(D^{t_i}) = POS_C(D^{t_i})$  then
15      $red(D^{t_j}) \leftarrow red(D^{t_j}) - \{c\}$ ;
16  $red(D^{t_i}) \leftarrow red(D^{t_j})$ ;
17 return  $red(D^{t_i})$ ;
```

Figure 3 illustrates the reduction process from the coarse to the fine label level. It contains three parts: the input of an HDS and $red(D^{t_j})$, the search for discernibility attributes, and the removal of redundant attributes to output $red(D^{t_i})$. In the middle part, two positive regions are computed, and only their difference is POS_{diff}^{cf} is considered. For each $s_u \in POS_{diff}^{cf}$, we find the $object^{cf}(s_u)$ that are in the same coarse label level decision class as s_u , in different fine label level decision classes, and indiscernible with respect to $red(D^{t_j})$. For each $s_u \in POS_{diff}^{cf}$ and each $s_v \in object^{cf}(s_u)$, the discernibility attribute $att^{cf}(s_u, s_v)$ is computed. The attributes that can distinguish these object pairs are taken as the discernibility attributes att^{cf} and added to $red(D^{t_j})$. After removing redundant attributes, $red(D^{t_i})$ is obtained.

Next, Example 4 illustrates the detailed process of deriving the fine label level reduct from the coarse label level reduct based on Algorithm 1.

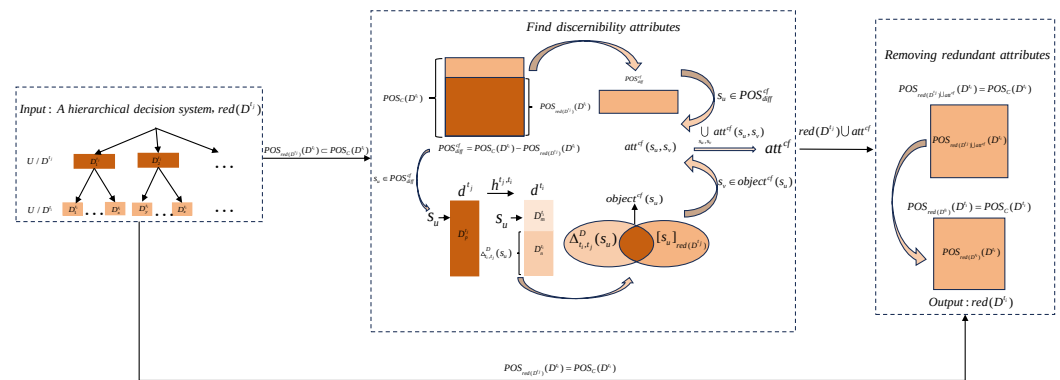


Figure 3. Attribute reduction from the coarse to the fine label level.

Example 4. According to Example 1, we can obtain the $red(D^2) = \{\{c_2, c_3\}, \{c_2, c_4\}\}$. By Algorithm 1, $red(D^1)$ can be calculated as follows:

(1) When $red(D^2) = \{c_2, c_3\}$.

Line 1: $POS_C(D^1) = \{s_2, s_4, s_7\}$, $POS_{red(D^2)}(D^1) = \{s_2, s_4\}$,

Lines 2–12: $POS_{diff}^f(D^1) = POS_C(D^1) - POS_{red(D^2)}(D^1) = \{s_7\}$, $att^{cf} = \emptyset$.

$s_7 \in POS_{diff}^f$, $\Delta_{1,2}^D(s_7) = [s_7]_{D^2} - [s_7]_{D^1} = \{s_1, s_4\}$, $[s_7]_{red(D^2)} = \{s_1, s_6, s_7\}$, $object^{cf}(s_7) = \Delta_{1,2}^D(s_7) \cap [s_7]_{red(D^2)} = \{s_1, s_4\} \cap \{s_1, s_6, s_7\} = \{s_1\}$, $c_1(s_7) \neq c_1(s_1)$. Accordingly, $att^{cf} = \{c_1\}$, $red(D^2) = att^{cf} \cup red(D^2) = \{c_1, c_2, c_3\}$.

Lines 13–15: $POS_{\{c_2, c_3\}}(D^1) \neq POS_C(D^1)$, $red(D^2) = \{c_1, c_2, c_3\}$;

$POS_{\{c_1, c_3\}}(D^1) = POS_C(D^1)$, $red(D^2) = \{c_1, c_3\}$;

$POS_{\{c_1\}}(D^1) \neq POS_C(D^1)$, $red(D^2) = \{c_1, c_3\}$.

So, the redundant attribute set is $\{c_2\}$.

Lines 16–17: $red(D^1) = red(D^2) = \{c_1, c_3\}$, return the reduct $red(D^1) = \{c_1, c_3\}$.

(2) Similarly, when $red(D^2) = \{c_2, c_4\}$, The subsequent process of computing att^{cf} is similar to the first case and will not be elaborated further. $att^{cf} = \{c_1, c_3\}$, $att^{cf} \cup red(D^2)$, removing redundancies, $red(D^1) = \{c_1, c_3\}$.

4.2. An Attribute Reduction Algorithm from the Fine to the Coarse Label Level

Based on the reduction relationship from fine to coarse provided in Theorem 2, Algorithm 2 is proposed, which mainly consists of the following parts. First, the positive region of the coarse decision attribute set with respect to the fine label level reduct and the condition attribute set is computed. Then, the algorithm compares the two positive regions. If they are not equal, their difference is computed. Otherwise, the coarse label level reduct can be directly obtained from the fine label level reduct by removing redundant attributes. The algorithm scans each object in this set. For each such object, the set of objects that need to be distinguished from it is computed according to Definition 11. Then, the object is compared with each object in this set, and a discernibility attribute that can distinguish the two objects is selected from the attributes not included in the fine label level reduct and added to the fine label level reduct. After that, redundant attributes are removed, and the reduct at the coarse label level is returned.

The time complexity of Algorithm 2 is $O(|U| \cdot |C| + |U'| \cdot |U''| \cdot |C'| + |U| \cdot |C''|^2)$, where $U' = POS_{diff}^f$, $U'' = object^{fc}(s_u)$, $C' = C - red(D^i)$, and $C'' = red(D^i)$. Here, C'' denotes the current reduct, which is initialized as $red(D^i)$ and may be updated by adding the selected discernibility attributes.

Algorithm 2: An attribute reduction algorithm from the fine to the coarse label level (LTFC-PR).

Input: A hierarchical decision system $HDS = (U, C \cup \{d^t | t = 1, 2, \dots, n\}, V, f)$; a reduct $red(D^{t_i})$.

Output: A reduct $red(D^{t_j})$.

- 1 Compute $POS_C(D^{t_j}), POS_{red(D^{t_i})}(D^{t_j})$;
- 2 **if** $POS_{red(D^{t_i})}(D^{t_j}) \neq POS_C(D^{t_j})$ **then**
- 3 $POS_{diff}^{fc} \leftarrow POS_C(D^{t_j}) - POS_{red(D^{t_i})}(D^{t_j})$;
- 4 Let $att^{fc} = \emptyset$;
- 5 **foreach** $s_u \in POS_{diff}^{fc}$ **do**
- 6 Calculate $object^{fc}(s_u)$ by Definition 11;
- 7 **foreach** $s_v \in object^{fc}(s_u)$ **do**
- 8 **foreach** $c \in C - red(D^{t_i})$ **do**
- 9 **if** $c(s_u) \neq c(s_v)$ **then**
- 10 $att^{fc} = att^{fc} \cup \{c\}$;
- 11 **break**;
- 12 $red(D^{t_i}) = red(D^{t_i}) \cup att^{fc}$;
- 13 **foreach** $c \in red(D^{t_i})$ **do**
- 14 **if** $POS_{red(D^{t_i})-\{c\}}(D^{t_j}) = POS_{red(D^{t_i})}(D^{t_j})$ **then**
- 15 $red(D^{t_i}) = red(D^{t_i}) - \{c\}$;
- 16 $red(D^{t_j}) = red(D^{t_i})$;
- 17 **return** $red(D^{t_j})$;

Figure 4 illustrates the reduction process from the fine to the coarse label level. It contains three parts: the input of an HDS and $red(D^{t_i})$, the search for discernibility attributes, and the removal of redundant attributes to output $red(D^{t_j})$. In the middle part, two positive regions are computed, and only their difference is POS_{diff}^{fc} is considered. For each $s_u \in POS_{diff}^{fc}$, we find the set of objects $object^{fc}(s_u)$ that do not belong to the positive region at the fine label level, belong to different decision classes from s_u at the coarse label level, and are indiscernible with respect to $red(D^{t_i})$. For each $s_u \in POS_{diff}^{fc}$ and each $s_v \in object^{fc}(s_u)$, the discernibility attribute $att^{fc}(s_u, s_v)$ is computed. The attributes that can distinguish these object pairs are taken as the discernibility attributes att^{fc} and added to $red(D^{t_i})$. After removing redundant attributes, $red(D^{t_j})$ is obtained.

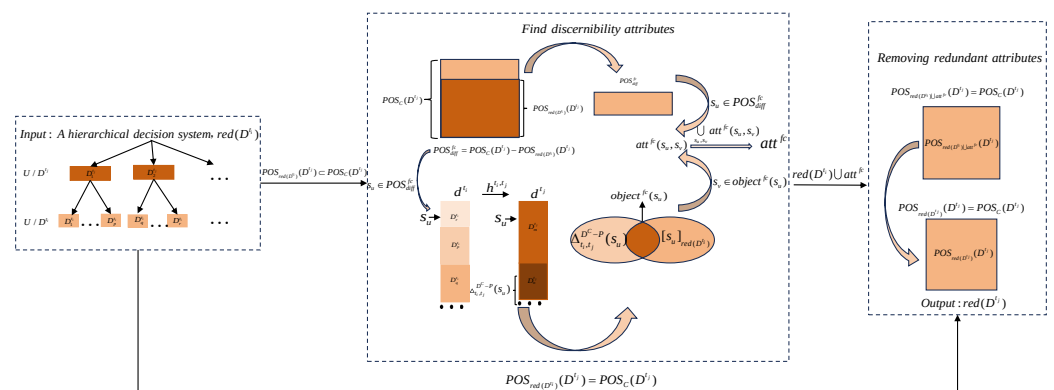


Figure 4. Attribute reduction from the fine to the coarse label level.

Next, Example 5 illustrates the detailed process of deriving the coarse label level reduct from the fine label level reduct based on Algorithm 2.

Example 5. According to Example 1, we can obtain the $red(D^1) = \{\{c_1, c_3\}, \{c_1, c_4, c_5\}\}$

(1) When $red(D^1) = \{c_1, c_3\}$.

Line 1: $POS_C(D^2) = \{s_1, s_2, s_4, s_6, s_7\}$, $POS_C(D^1) = \{s_2, s_4, s_7\}$,

Lines 2–12: $POS_{diff}^{fc} = POS_C(D^2) - POS_{red(D^1)}(D^2) = \{s_1, s_6\}$, $att^{fc} = \emptyset$. $s_1 \in POS_{diff}^{fc}$, $\Delta_{1,2}^{D^C-P}(s_1) = [s_1]_{D^2}^C - POS_C(D^1) = \{s_2, s_3, s_8, s_9, s_{10}\} - \{s_2, s_4, s_7\} = \{s_3, s_8, s_9, s_{10}\}$. So $object^{fc}(s_1) = \Delta_{1,2}^{D^C-P}(s_1) \cap [s_1]_{red(D^1)} = \{s_9, s_{10}\}$. When $s_9 \in object^{fc}(s_1)$, $c_2(s_1) \neq c_2(s_9)$; when $s_{10} \in object^{fc}(s_1)$, $c_2(s_1) \neq c_2(s_{10})$. So $att^{fc} = \{c_2\}$. Similarly, the discernibility attributes for s_6 can be obtained. Accordingly, $att^{fc} = \{c_2\}$. So, $red(D^1) = att^{fc} \cup red(D^1) = \{c_1, c_2, c_3\}$.

Lines 13–15: $POS_{\{c_2, c_3\}}(D^2) = POS_C(D^2)$, $red(D^1) = \{c_2, c_3\}$. $POS_{\{c_3\}}(D^2) \neq POS_C(D^2)$, $red(D^1) = \{c_2, c_3\}$. $POS_{\{c_2\}}(D^1) \neq POS_C(D^1)$, $red(D^1) = \{c_2, c_3\}$. So, the redundant attribute set is $\{c_1\}$.

Lines 16–17: $red(D^2) = red(D^1)$, removing redundancy, the $red(D^2) = \{c_2, c_3\}$.

(2) When $red(D^1) = \{c_1, c_4, c_5\}$. The subsequent process of computing att^{fc} is similar to the first case and will not be elaborated further. Accordingly, $att^{fc} = \{c_2\}$. So, $att^{fc} \cup red(D^1) = \{c_1, c_2, c_4, c_5\}$, removing redundancies, the $red(D^2) = \{c_2, c_4\}$.

The algorithms LTCF-PR and LTFC-PR derive the reduct at one label level from the reduct at another label level. Instead of computing core attributes as in many reduction algorithms, this study computes discernibility attributes and makes use of the available information. As a result, reduct derivation across different label levels is achieved without restarting heuristic search at each level. In this way, the proposed methods reduce the search space, reduce repeated computation, and improve computational efficiency.

5. Experimental Analyses

To evaluate Algorithms 1 and 2, we compare them with four attribute reduction algorithms: PR [42], FSPA-PR [17], AR-CT-PR [45], and GS3WR [9]. The experimental datasets are obtained from the UCI (<https://archive.ics.uci.edu>, accessed on 15 December 2024). The detailed information on the datasets is listed in Table 2. In this paper, $|U|$ is the cardinality of the universe, $|C|$ is the cardinality of the condition attribute set, $|V_{d^1}|$ refers to the number of decision classes at the fine label level ($t = 1$), and $|V_{d^2}|$ refers to the number of decision classes at the coarse label level ($t = 2$). The datasets include different data types, and the same preprocessing method is used. Specifically, equal-frequency discretization in WEKA 3.6 is used for continuous data, and missing values are filled using the most frequent attribute value. All experiments were conducted on a personal computer running Microsoft Windows 10 with an Intel Core i5 processor and 8 GB RAM. The algorithms were implemented in Python 3.9.13, and PyCharm 2024.3.6 was adopted as the software platform.

Table 2. UCI Datasets.

No.	Dataset	Abbr.	$ U $	$ C $	$ V_{d^1} $	$ V_{d^2} $
1	Audiology	Aud	200	69	24	3
2	Dermatology	Der	366	34	6	2
3	Zoo	Zoo	101	16	7	3
4	Forest Type Mapping	For	198	27	4	2
5	Waveform Database Generator	Wav	5000	40	3	2
6	Absenteeism at work	Abs	740	20	19	3
7	Lymphography	Lym	148	19	3	2
8	Cardiotocography	Car	2126	21	10	3
9	Arrhythmia	Arr	452	279	16	3
10	Semeion	Sem	1593	256	10	3
11	Urban Land Cover	Urb	675	147	9	3
12	ISOLET	ISO	7797	617	26	3

In the experiments, all datasets were organized into two label levels by hierarchical clustering, where t denotes the label level, where smaller values of t denote finer label levels and larger values denote coarser label levels. Therefore, $t = 1$ and $t = 2$ were selected to represent the fine and coarse label levels, respectively. The two label-level setting is sufficient to evaluate the reduction relationships between different label levels and the performance of the algorithms. An attribute reduction algorithm from the coarse to the fine label level is abbreviated as LTCF-PR. An attribute reduction algorithm from the fine to the coarse label level is abbreviated as LTFC-PR.

5.1. Experimental Analyses of LTCF-PR

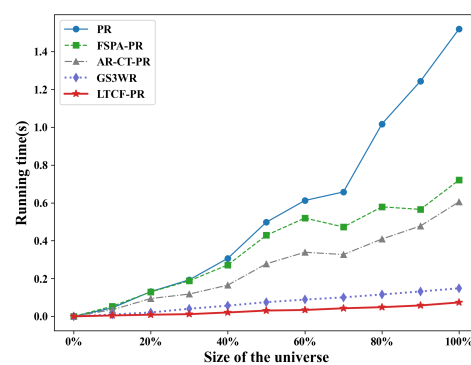
To evaluate the efficiency of LTCF-PR, LTCF-PR is compared with the other four attribute reduction algorithms that adopt positive region consistency as the reduction objective. Since all algorithms use the same method for equivalence class partitions, we ignore differences in computational efficiency due to equivalence class partitions. For the other four algorithms, computing core attributes is a crucial step. Due to the adoption of computing discernibility attributes instead of computing core attributes, LTCF-PR has higher efficiency.

Table 3 presents the running time on eight datasets. LTCF-PR has the shortest running time on all datasets. For example, on the Aud dataset, LTCF-PR is about 20.6, 9.8, 8.2, and 2.0 times faster than the other algorithms, respectively. On the dataset Wav, PR takes 40.5537 s, while LTCF-PR takes 0.5831 s. Thus, LTCF-PR is about 69.6 times faster than PR. LTCF-PR is about 26.3 and 20.6 times faster than FSPA-PR and AR-CT-PR, and about 7.8 times faster than GS3WR. Similar results are found on the datasets Der, Zoo, For, Abs, Lym, and Car. On some datasets with a large number of features, LTCF-PR still has the shortest running time. In particular, on dataset ISO, LTCF-PR only takes 0.9494 s, while the other four algorithms take 1796.5967 s, 1573.3061 s, 1065.8574 s, and 70.9131 s, respectively. As a whole, LTCF-PR has the shortest average time of 0.1715 s. The bold values in each row of the table indicate the algorithm with the fastest running speed in that row.

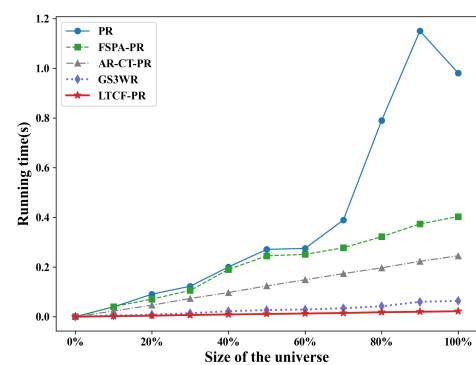
Table 3. Comparison of running time(s).

No.	Abbr.	PR	FSPA-PR	AR-CT-PR	GS3WR	LTCF-PR
1	Aud	1.5192	0.7207	0.6059	0.1468	0.0738
2	Der	0.9808	0.4031	0.2452	0.0631	0.0219
3	Zoo	0.0641	0.0331	0.0139	0.0112	0.0036
4	For	0.1348	0.1037	0.0883	0.0269	0.0069
5	Wav	40.5537	15.3261	12.0362	4.5276	0.5831
6	Abs	0.3956	0.3392	0.2981	0.0088	0.0056
7	Lym	0.0819	0.0478	0.0428	0.0139	0.0019
8	Car	0.6478	0.5992	0.4491	0.3099	0.1432
9	Arr	14.3016	14.4743	11.1655	0.6133	0.0141
10	Sem	93.5695	58.5012	62.7000	5.6637	0.2393
11	Urb	9.2893	6.9526	5.7597	0.5050	0.0150
12	ISO	1796.5967	1573.3061	1065.8574	70.9131	0.9494
	Avg	163.1780	139.2340	96.6052	6.9003	0.1715

Figure 5 illustrates the running time trends of the five algorithms as the size of the universe increases. The x-axis represents the universe size, which is divided into ten parts from 0% to 100%. The y-axis represents the running time of the algorithms. As the universe size increases, the running times of the five algorithms increase. However, their increases are clearly different. In all subfigures, LTCF-PR shows the slowest increase in running time. In most subfigures, LTCF-PR's curve is nearly parallel to the x-axis, even when the universe size becomes large. By comparison, FSPA-PR, AR-CT-PR, and GS3WR show a greater increase as the size of the universe increases. Taking Figure 5a as an example. When the universe size increases from 0% to 100%, the running time of PR increases from 0 to about 1.52 s. At the same point, the running times of FSPA-PR and AR-CT-PR are about 0.72 s and 0.61 s, and GS3WR is about 0.15 s. By comparison, the running time of LTCF-PR is only about 0.074 s, which is much less than the others. A similar result can be obtained in Figure 5b. When the universe size reaches 100%, PR takes about 0.98 s, while LTCF-PR takes only about 0.022 s. It should be noted that the running time is not strictly monotonic in some subfigures. For example, in Figure 5c, the running time of PR decreases from about 0.048 s at 50% to 0.033 s at 60%. In this case, the running time of LTCF-PR remains stable, and its variation is much smaller than that of the other algorithms. Figure 5a–l show that LTCF-PR has the shortest running time for the same universe size in most cases. Its advantage becomes more obvious as the universe size increases. These results show that LTCF-PR has higher time efficiency than the other four algorithms.

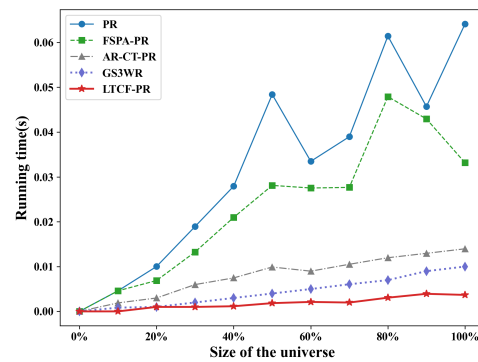


(a) Aud

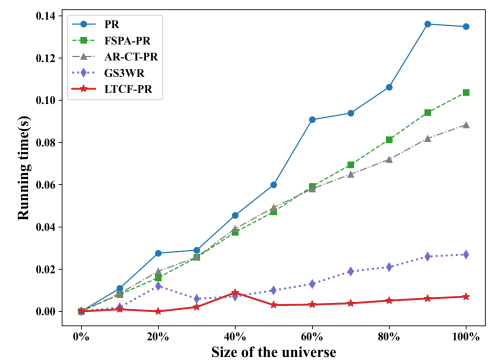


(b) Der

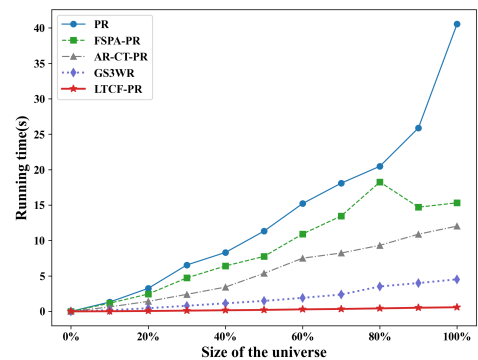
Figure 5. Cont.



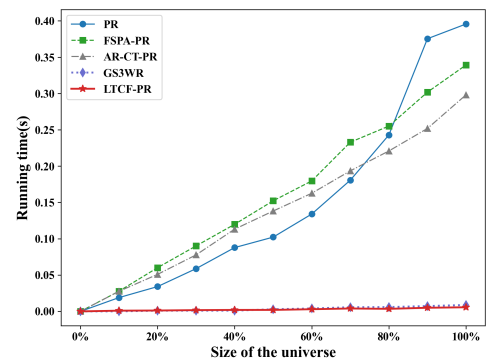
(c) Zoo



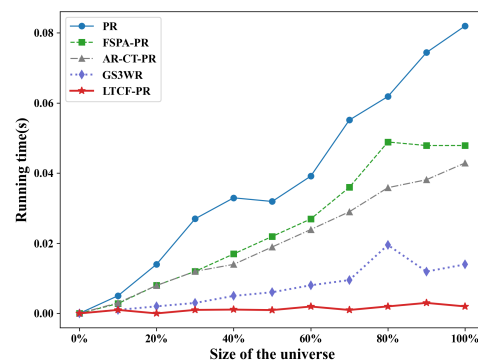
(d) For



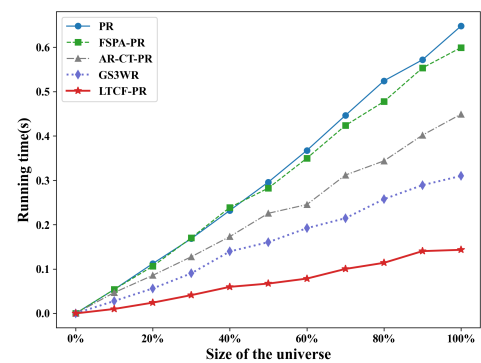
(e) Wav



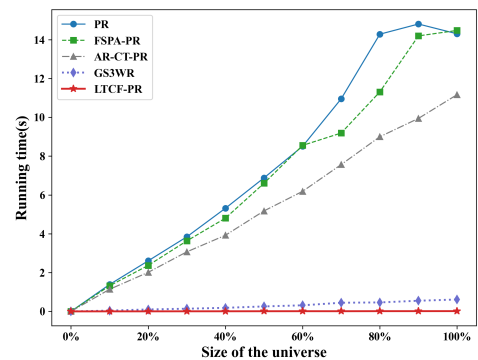
(f) Abs



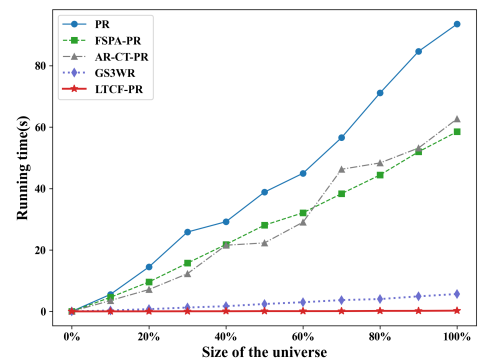
(g) Lym



(h) Car



(i) Arr



(j) Sem

Figure 5. Cont.

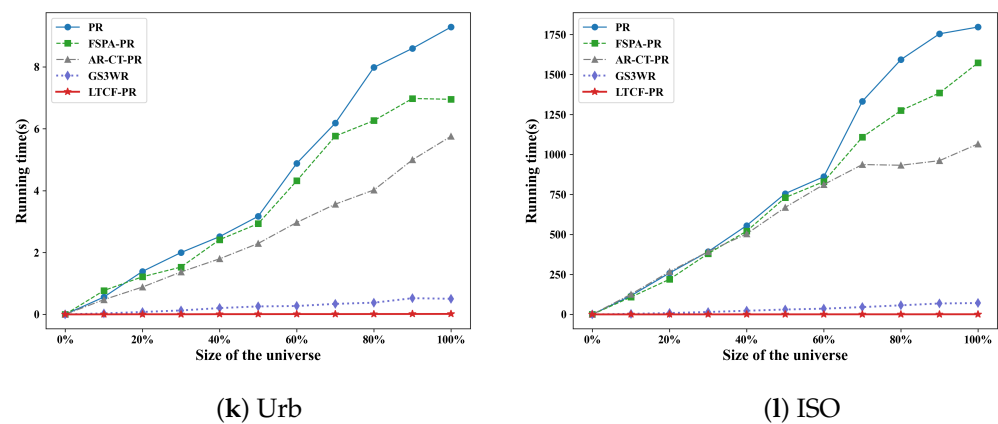


Figure 5. Comparison of four reduction algorithms and LTCF-PR as the universe size increases.

Table 4 presents the reduction results of PR and LTCF-PR on different datasets. Since the other four algorithms obtain the same reduction result, Table 4 presents only the reduction results obtained by PR. The algorithms LTCF-PR and PR either have the same or very similar reduction results. On the datasets Zoo, Abs, Lym, and Car, PR and LTCF-PR obtain the same reduction results. For example, on the dataset Lym, the reduct obtained by PR and LTCF-PR is {2, 13, 14, 15, 16, 18}. For the datasets Aud, Der, and Wav, LTCF-PR obtains reduction results that differ in a small number of attributes. For example, on the dataset Aud, the reduction result obtained by PR is {1, 2, 4, 5, 6, 7, 11, 13, 14, 15, 36, 47, 64, 65, 66}, while the reduction result obtained by LTCF-PR is {1, 2, 5, 6, 7, 8, 10, 11, 15, 47, 59, 62, 64, 66, 69}. Although the reduction results are not exactly the same, both reduction results have the same length. For high-dimensional datasets with many features, such as Arr, Sem, Urb, and ISO, the reducts obtained by LTCF-PR and PR differ only slightly in the length of the reduct.

Table 4. Comparison of reduction results between PR and LTCF-PR.

No.	Abbr.	Algorithms	Reduction Results
1	Aud	PR	{1, 2, 4, 5, 6, 7, 11, 13, 14, 15, 36, 47, 64, 65, 66}
		LTCF-PR	{1, 2, 5, 6, 7, 8, 10, 11, 15, 47, 59, 62, 64, 66, 69}
2	Der	PR	{1, 5, 14, 15, 22, 26, 27, 28, 34}
		LTCF-PR	{1, 5, 14, 15, 22, 26, 28, 33, 34}
3	Zoo	PR	{3, 4, 6, 8, 13}
		LTCF-PR	{3, 4, 6, 8, 13}
4	For	PR	{1, 3, 16, 17, 26}
		LTCF-PR	{7, 9, 10, 11, 20}
5	Wav	PR	{1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12, 13, 15, 16, 18, 20, 24, 26, 30, 33, 37, 38, 39, 40}
		LTCF-PR	{3, 4, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 24, 29, 31, 32, 38, 39, 40}
6	Abs	PR	{1, 2, 4, 5, 10}
		LTCF-PR	{1, 2, 4, 5, 10}
7	Lym	PR	{2, 13, 14, 15, 16, 18}
		LTCF-PR	{2, 13, 14, 15, 16, 18}
8	Car	PR	{1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 21}
		LTCF-PR	{1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 21}
9	Arr	PR	{8, 15, 17, 117, 170, 179, 248}
		LTCF-PR	{1, 15, 57, 126, 137, 187, 240}
10	Sem	PR	{9, 13, 50, 52, 54, 58, 62, 87, 108, 112, 135, 140, ..., 184, 196, 198, 221, 247, 248}
		LTCF-PR	{1, 2, 6, 10, 13, 16, 25, 37, 42, 75, 78, 84, 88, 111, ..., 158, 178, 221, 226, 236, 255}
11	Urb	PR	{5, 19, 46, 63, 64, 65, 102}
		LTCF-PR	{19, 20, 26, 114, 125, 143, 144}
12	ISO	PR	{4, 15, 88, 145, 304, 305, 331, 376, 404, 481, 600}
		LTCF-PR	{1, 5, 6, 16, 209, 214, 282, 396, 433, 560, 602, 608}

Table 5 presents the classification accuracy based on Support Vector Machine (SVM) and K-Nearest Neighbors (KNN) classifiers. The classification accuracy is evaluated using the 10-fold cross-validation method. For the SVM classifier, LTFC-PR has slightly higher classification accuracy than PR on the datasets Der, Zoo, Sem, and ISO. On the datasets Zoo, Abs, Lym, and Car, PR and LTFC-PR obtain the same classification accuracy. For the KNN classifier, LTFC-PR has higher classification accuracy than PR on the datasets Der, For, Wav, Sem, Urb, and ISO. On the datasets Zoo, Abs, Lym, and Car, PR, and LTFC-PR have the same classification accuracy. In Table 5, the values with higher classification accuracy are shown in bold.

Table 5. Comparison of classification accuracy based on SVM and KNN (%).

No.	Abbr.	SVM		KNN	
		PR	LTFC-PR	PR	LTFC-PR
1	Aud	0.7400 $\pm$ 0.1261	0.7150 $\pm$ 0.1074	0.6950 $\pm$ 0.1491	0.6850 $\pm$ 0.1305
2	Der	0.9402 $\pm$ 0.0395	0.9429 $\pm$ 0.0370	0.9510 $\pm$ 0.0314	0.9564 $\pm$ 0.0324
3	Zoo	0.8409 $\pm$ 0.0672	0.8409 $\pm$ 0.0672	0.8818 $\pm$ 0.0966	0.8818 $\pm$ 0.0966
4	For	0.9450 $\pm$ 0.0610	0.9447 $\pm$ 0.0350	0.9395 $\pm$ 0.0441	0.9500 $\pm$ 0.0316
5	Wav	0.7445 $\pm$ 0.0165	0.7427 $\pm$ 0.0231	0.6811 $\pm$ 0.0176	0.6995 $\pm$ 0.0207
6	Abs	0.3932 $\pm$ 0.0562	0.3932 $\pm$ 0.0562	0.3189 $\pm$ 0.0786	0.3189 $\pm$ 0.0786
7	Lym	0.7357 $\pm$ 0.0892	0.7357 $\pm$ 0.0892	0.7562 $\pm$ 0.1054	0.7562 $\pm$ 0.1054
8	Car	0.5428 $\pm$ 0.0454	0.5428 $\pm$ 0.0454	0.4770 $\pm$ 0.0488	0.4770 $\pm$ 0.0488
9	Arr	0.5598 $\pm$ 0.0245	0.5398 $\pm$ 0.0133	0.5510 $\pm$ 0.0434	0.5131 $\pm$ 0.0351
10	Sem	0.7025 $\pm$ 0.0233	0.7602 $\pm$ 0.0329	0.6930 $\pm$ 0.0368	0.7182 $\pm$ 0.0227
11	Urb	0.7556 $\pm$ 0.0428	0.7540 $\pm$ 0.0328	0.6917 $\pm$ 0.0584	0.6975 $\pm$ 0.0543
12	ISO	0.4430 $\pm$ 0.0127	0.5838 $\pm$ 0.0165	0.3799 $\pm$ 0.0118	0.4880 $\pm$ 0.0151

5.2. Experimental Analyses of LTFC-PR

To evaluate the efficiency of LTFC-PR, LTFC-PR is compared with four attribute reduction algorithms that adopt a positive region consistency as the reduction objective.

Table 6 presents the running time comparison of five algorithms on eight datasets. LTFC-PR has the shortest running time on all datasets. From Table 6, LTFC-PR is faster than the other four algorithms. For example, on the dataset Der, LTFC-PR is about 21.3 times faster than PR, and about 13.1, 8.2, and 2.4 times faster than FSPA-PR, AR-CT-PR, and GS3WR, respectively. A similar result can be found on the large dataset Wav. PR takes 28.5217 s, whereas LTFC-PR takes only 0.5192 s. In this case, LTFC-PR is about 55 times faster than PR. LTFC-PR is about 23, 16, and 6 times faster than FSPA-PR, AR-CT-PR, and GS3WR, respectively. On smaller datasets such as Zoo, For, Abs, and Lym, LTFC-PR consistently has the shortest running time. On datasets with a large number of features, such as Arr, Sem, Urb, and ISO, LTFC-PR still has the shortest running time. In particular, on ISO, LTFC-PR takes 1.0635 s, whereas PR, FSPA-PR, AR-CT-PR, and GS3WR take 1803.1283 s, 1411.3298 s, 1352.3382 s, and 87.2543 s, respectively. Similar results are also observed on Arr, Sem, and Urb. LTFC-PR has the shortest average running time of 0.1675 s. LTFC-PR is about 975, 621, 716, and 48 times faster than the other four algorithms, respectively. The bold values in each row of the table indicate the algorithm with the fastest running speed in that row.

Table 6. Comparison of running time(s).

No.	Abbr.	PR	FSPA-PR	AR-CT-PR	GS3WR	LTFC-PR
1	Aud	2.8742	1.9022	0.7269	0.0367	0.0247
2	Der	0.5729	0.3532	0.2216	0.0644	0.0269
3	Zoo	0.0648	0.0349	0.0131	0.0079	0.0029
4	For	0.1152	0.0941	0.0801	0.0182	0.0062
5	Wav	28.5217	11.9942	8.1415	3.2322	0.5192
6	Abs	0.2881	0.2287	0.2793	0.0061	0.0045
7	Lym	0.0694	0.0469	0.0431	0.0119	0.0021
8	Car	0.6074	0.5742	0.3952	0.2992	0.1307
9	Arr	18.6312	16.2919	14.0990	0.6353	0.0180
10	Sem	96.7593	69.1702	58.5133	5.1481	0.1871
11	Urb	9.4473	7.6586	6.5824	0.4857	0.0245
12	ISO	1803.1283	1141.3298	1352.3382	87.2543	1.0635
	Avg	163.4233	104.1399	120.1195	8.1000	0.1675

Figure 6 illustrates the running time trends of PR, FSPA-PR, AR-CT-PR, GS3WR, and LTFC-PR with the size of the universe increasing. The x-axis represents the universe size. The y-axis represents the running time. The running times of the other four algorithms grow rapidly. In contrast, LTFC-PR has the shortest running time on all subfigures and shows the slowest increasing trend. For example, in Figure 6e, the running time of LTFC-PR increases very slowly as the universe size increases. Its curve is almost parallel to the x-axis for the whole range. In contrast, the curves of PR, FSPA-PR, AR-CT-PR, and GS3WR show much larger slopes, and their running times increase rapidly with the universe size. It should be noted that the running time is not strictly monotonic in some subfigures. Taking Figure 6g as an example, the curve of LTFC-PR remains very smooth over the whole range of universe sizes. From 0% to 100%, its running time changes only slightly as the universe size increases. In contrast, the curves of PR, FSPA-PR, and AR-CT-PR show larger variations, especially from 40% to 80%. GS3WR also shows visible changes in this range. A similar case can also be shown in Figure 6a. The curve of LTFC-PR remains very smooth over the whole range of universe sizes, and its running time changes only slightly from 0% to 100%. In contrast, the curves of PR, FSPA-PR, and AR-CT-PR show larger variations. On all subfigures, LTFC-PR consistently ranks first in time efficiency. Figure 6a–l shows that LTFC-PR has the shortest running time for the same universe size in most cases. Its advantage becomes more obvious as the universe size increases. These results show that LTFC-PR has higher time efficiency than the other four algorithms.

Table 7 presents the reduction results of PR and LTFC-PR on different datasets. Since the other four attribute reduction algorithms obtain the same reducts on all datasets, Table 4 presents only the reducts obtained by PR. The algorithms LTFC-PR and PR either have the same or very similar reduction results. On the datasets Abs, Lym, Car, PR, and LTFC-PR we obtain the same reduction results. For example, on dataset Abs, PR and LTFC-PR obtain the reduct {1, 2, 4, 5, 10}. On dataset Lym, PR and LTFC-PR obtain the reduction result {2, 13, 14, 15, 16, 18}. A similar case is shown in the dataset Car, where PR and LTFC-PR obtain the same reduction results. On the datasets Aud, Der, and Zoo, the reduction results obtained by LTFC-PR are not exactly the same as those obtained by PR, but they are highly similar in length. For instance, on the dataset Aud, PR obtains the reduction result {2, 6, 7, 8, 11, 37, 58, 59, 62, 64, 66, 69}, while LTFC-PR obtains {2, 4, 6, 7, 11, 13, 14, 15, 64, 65, 66}. For high-dimensional datasets with many features, such

as Arr, Sem, Urb, and ISO, the reducts obtained by LTFC-PR and PR differ only slightly in the length of the reduct.

Table 8 presents the classification accuracy based on Support Vector Machine (SVM) and K-Nearest Neighbors (KNN) classifiers. The classification accuracy is evaluated using the 10-fold cross-validation method. For the SVM classifier, LTFC-PR has slightly higher classification accuracy than PR on the datasets For, Wav, Sem, and Urb. On the datasets Abs, Lym, Car, and Arr, PR and LTFC-PR have the same classification accuracy. For the KNN classifier, LTFC-PR has slightly higher classification accuracy than PR on the datasets Aud, Zoo, Arr, and Sem. On the datasets Abs, Lym, and Car, PR and LTFC-PR have the same classification accuracy. In Table 8, the values with higher classification accuracy are shown in bold.

Table 9 presents the number of rules for two label levels, $t = 1$ and $t = 2$, on the different datasets. In general, the number of rules for the coarse label level ($t = 2$) is less than or equal to that for the fine label level ($t = 1$) on most datasets. Specifically, the number of rules is reduced from 143 to 106 on the dataset Aud, from 329 to 327 on the dataset Der, from 147 to 119 on the dataset For, from 706 to 680 on the dataset Abs, and from 1391 to 1282 on the dataset Car. These results suggest that using a coarser label level can lead to more compact rule sets, which may help alleviate rule redundancy while preserving the overall descriptive ability of the rules. The bold values indicate that fewer decision rules can be obtained at the corresponding label level.

Table 7. Comparison of reduction results between PR and LTFC-PR.

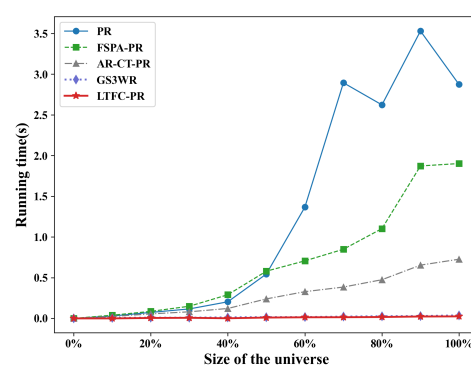
No.	Abbr.	Algorithms	Reduction Results
1	Aud	PR	{2, 6, 7, 8, 11, 37, 58, 59, 62, 64, 66, 69}
		LTFC-PR	{2, 4, 6, 7, 11, 13, 14, 15, 64, 65, 66}
2	Der	PR	{1, 5, 14, 15, 22, 26, 28, 33, 34}
		LTFC-PR	{1, 5, 14, 15, 22, 26, 27, 28, 34}
3	Zoo	PR	{1, 3, 6, 8, 13}
		LTFC-PR	{3, 4, 6, 8, 13}
4	For	PR	{7, 10, 11, 20}
		LTFC-PR	{1, 3, 16, 17, 26}
5	Wav	PR	{3, 4, 5, 6, 7, 9, 10, 11, 12, 13, 14, 16, 17, 18, 19, 24, 29, 31, 32, 38, 39, 40}
		LTFC-PR	{1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12, 13, 15, 16, 18, 20, 24, 26, 30, 33, 37, 38, 39, 40}
6	Abs	PR	{1, 2, 4, 5, 10}
		LTFC-PR	{1, 2, 4, 5, 10}
7	Lym	PR	{2, 13, 14, 15, 16, 18}
		LTFC-PR	{2, 13, 14, 15, 16, 18}
8	Car	PR	{1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 21}
		LTFC-PR	{1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 21}
9	Arr	PR	{15, 57, 126, 137, 187, 240}
		LTFC-PR	{8, 15, 17, 117, 170, 179, 248}
10	Sem	PR	{10, 13, 16, 25, 37, 42, 75, 78, 84, 88, 111, 114, . . . , 178, 221, 226, 236, 255}
		LTFC-PR	{9, 13, 50, 52, 54, 58, 62, 87, 108, 112, 135, 140, . . . , 184, 196, 198, 221, 248}
11	Urb	PR	{19, 20, 26, 114, 125, 143, 144}
		LTFC-PR	{5, 19, 46, 63, 64, 65, 102}
12	ISO	PR	{5, 69, 209, 214, 282, 396, 433, 560, 602, 608}
		LTFC-PR	{1, 5, 6, 16, 209, 214, 282, 396, 433, 560, 602, 608}

Table 8. Comparison of classification accuracy based on SVM and KNN (%).

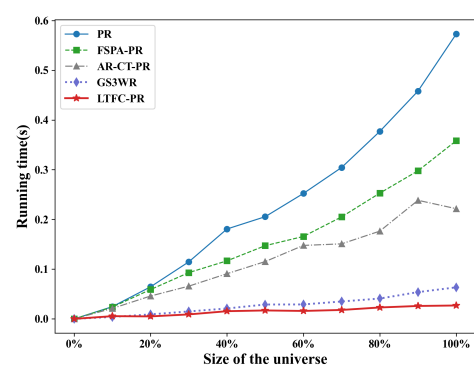
No.	Abbr.	SVM		KNN	
		PR	LTFC-PR	PR	LTFC-PR
1	Aud	0.8600 $\pm$ 0.0624	0.8300 $\pm$ 0.0872	0.8300 $\pm$ 0.0678	0.8700 $\pm$ 0.0714
2	Der	0.9292 $\pm$ 0.0453	0.9211 $\pm$ 0.0474	0.9671 $\pm$ 0.0166	0.9645 $\pm$ 0.0174
3	Zoo	0.9118 $\pm$ 0.1026	0.9018 $\pm$ 0.1169	0.8905 $\pm$ 0.1782	0.9009 $\pm$ 0.1184
4	For	0.9139 $\pm$ 0.0711	0.9500 $\pm$ 0.0548	0.9650 $\pm$ 0.0391	0.9650 $\pm$ 0.0391
5	Wav	0.8410 $\pm$ 0.0114	0.8420 $\pm$ 0.0120	0.8206 $\pm$ 0.0113	0.8128 $\pm$ 0.0100
6	Abs	0.9149 $\pm$ 0.0062	0.9149 $\pm$ 0.0062	0.8905 $\pm$ 0.0542	0.8905 $\pm$ 0.0542
7	Lym	0.8100 $\pm$ 0.0968	0.8100 $\pm$ 0.0968	0.8167 $\pm$ 0.0871	0.8167 $\pm$ 0.0871
8	Car	0.7230 $\pm$ 0.0624	0.7230 $\pm$ 0.0624	0.7230 $\pm$ 0.0624	0.7230 $\pm$ 0.0624
9	Arr	0.7345 $\pm$ 0.0088	0.7345 $\pm$ 0.0088	0.7367 $\pm$ 0.0548	0.7389 $\pm$ 0.0516
10	Sem	0.7005 $\pm$ 0.0439	0.7162 $\pm$ 0.0325	0.7784 $\pm$ 0.0385	0.7878 $\pm$ 0.0384
11	Urb	0.6089 $\pm$ 0.0699	0.6533 $\pm$ 0.0622	0.7095 $\pm$ 0.0611	0.6875 $\pm$ 0.0568
12	ISO	0.6168 $\pm$ 0.0099	0.5382 $\pm$ 0.0185	0.6581 $\pm$ 0.0196	0.6005 $\pm$ 0.0195

Table 9. Comparison of the number of rules for different label levels.

No.	Abbr.	Number of Rules	
		t = 1	t = 2
1	Aud	143	106
2	Der	329	327
3	Zoo	22	22
4	For	147	119
5	Wav	4987	4987
6	Abs	706	680
7	Lym	129	129
8	Car	1391	1282
9	Arr	451	451
10	Sem	1462	1439
11	Urb	657	657
12	ISO	7788	7795

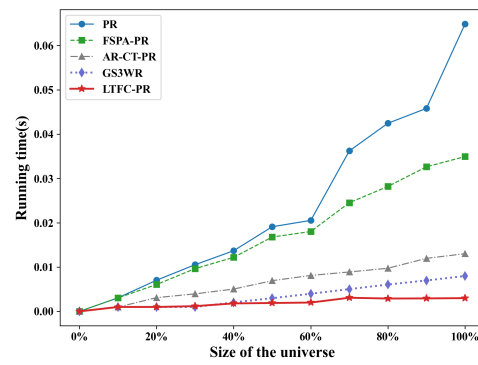


(a) Aud

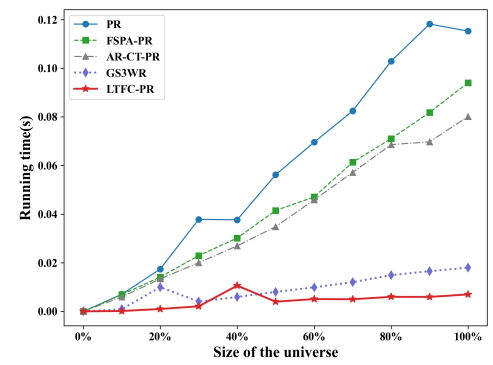


(b) Der

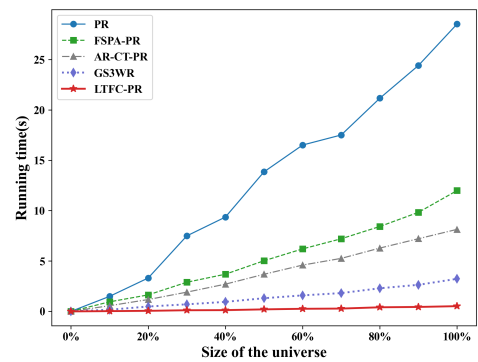
Figure 6. Cont.



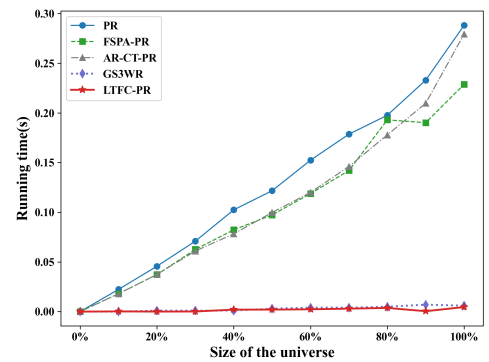
(c) Zoo



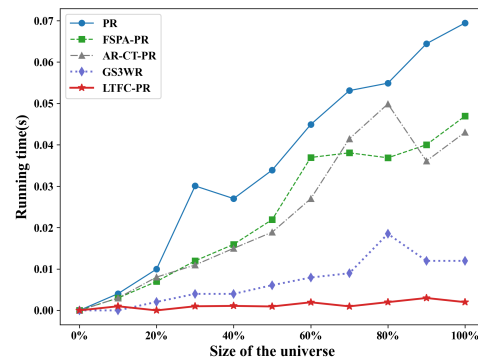
(d) For



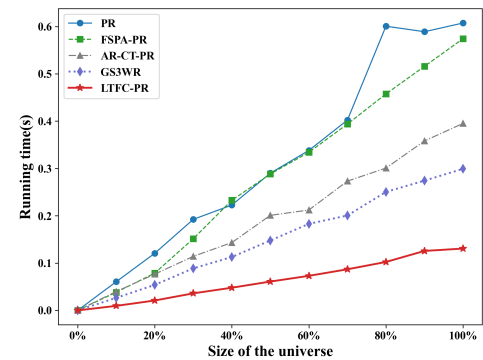
(e) Wav



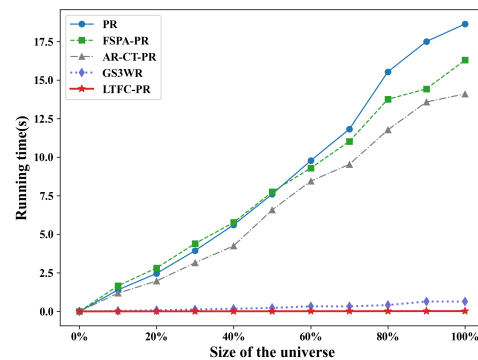
(f) Abs



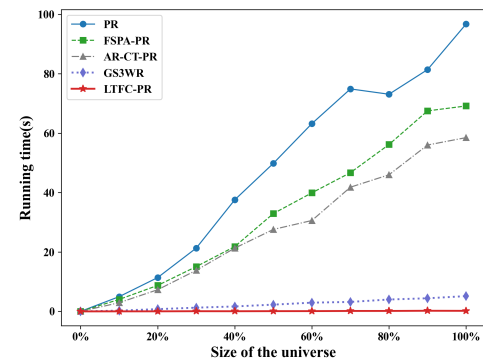
(g) Lym



(h) Car



(i) Arr



(j) Sem

Figure 6. Cont.

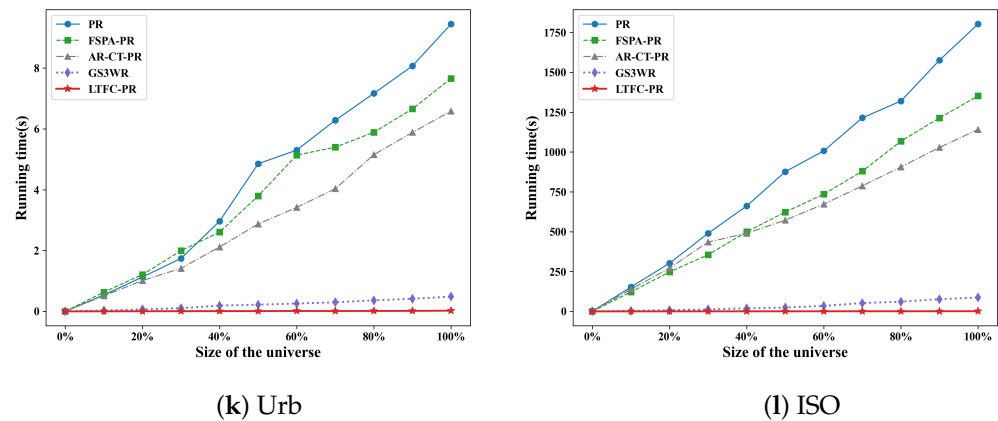


Figure 6. Comparison of four reduction algorithms and LTFC-PR as the universe size increases.

5.3. Limitations of Research

The limitations of this paper are summarized as follows.

(1) Although the proposed methods improve computational efficiency, they may still have limitations when dealing with very large-scale datasets. This is because, as the data size increases, the number of object pairs that need to be distinguished also increases, which leads to more computation time.

(2) This work does not explore extensions to non-classical rough set models, such as fuzzy rough sets, probabilistic rough sets, and incomplete decision systems, which limits the generality of the proposed methods to some extent.

6. Conclusions

This paper proposes a method for deriving the reduct at one label level from that at another label level. The main contribution of this paper is the establishment of reduction relationships between different label levels and the design of two efficient attribute reduction algorithms. Instead of computing core attributes as in many reduction algorithms, this study computes discernibility attributes and makes use of the available information. As a result, LTCF-PR and LTFC-PR reduce the number of iterations required for reduction, thereby significantly improving computational efficiency. Moreover, the established reduction relationships provide a new model for deriving the reduct from one label level to another in rough set theory and hierarchical learning, while the proposed algorithms provide an efficient way to reduce computational cost in hierarchical classification.

Future research will focus on several directions. (1) Further improving the efficiency of computing distinguishable object pairs, especially for large-scale datasets. (2) Investigating an incremental method for dynamic datasets, where objects or attributes may change over time. (3) Exploring the integration of the proposed methods with other uncertainty models, such as fuzzy rough sets and probabilistic rough sets, to better handle uncertainty in complex data. The proposed methods will be further extended to practical application scenarios with hierarchical structures, such as text classification, bioinformatics, and medical diagnosis, to address the attribute reduction problem in hierarchical classification and to further promote the practical development of RST.

Author Contributions: Conceptualization, R.W. and N.Z.; software, R.W.; data curation, R.W.; visualization, R.W. and N.Z.; writing—original draft preparation, R.W.; writing—review and editing, R.W., N.Z. and Y.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Natural Science Foundation of China under Grant No. 62273290, Smart City Innovation Lab Project under Grant No. SDGP202302000504.

Data Availability Statement: The original contributions presented in this study are included in the article. The code and datasets used in this study are available at the following link: https://github.com/sinanqingjiayou/LTCF_PR-LTFC_PR.git (accessed on 29 March 2026).

Acknowledgments: Thanks for the discussions with Jinming Gu during the development of the proposed method. In addition, we appreciate her valuable suggestions.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations and symbols are used in this manuscript:

DS	Decision system
LA	Lower approximation
UA	Upper approximation
HDS	Hierarchical decision system
$PRCS$	Positive region consistent set
d^t	Decision attribute at the t -th label level
D^t	Decision attribute set at the t -th label level
D_i^t	i -th decision class at the t -th label level
$red(D^{t_i})$	Reduct at the t_i -th label level
$red(D^{t_j})$	Reduct at the t_j -th label level
DM_t	Discernibility matrix at the t -th label level
$m_t(s_u, s_v)$	Discernibility attribute set between s_u and s_v at the t -th label level
R	Redundant attribute set
$\Delta_{t_i, t_j}^D(s_u)$	The set difference between decision classes
$R_{POS}^{t_i}$	The set of object pairs not discernible at the coarse label level
POS_{diff}^{cf}	The set of the positive region objects indistinguishable by the coarse label reduct
$object^{cf}(s_u)$	Objects that cannot be distinguished from s_u by the coarse label level reduct
$att^{cf}(s_u, s_v)$	A discernibility attribute for the object pair from coarse to fine label levels
att^{cf}	The discernibility attribute set from coarse to fine label levels
Δ_{t_i, t_j}^{POS}	The set difference between positive regions
$R_{POS}^{t_j}$	The set of object pairs not discernible at the fine label level
$\Delta_{t_i, t_j}^{D^c-P}(s_u)$	The set difference between decision classes and positive region
POS_{diff}^{fc}	The set of the positive region objects indistinguished by the fine label level reduct
$object^{fc}(s_u)$	Objects that cannot be distinguished from s_u by the fine label level reduct
$att^{fc}(s_u, s_v)$	A discernibility attribute for the object pair from fine to coarse label levels
att^{fc}	The discernibility attribute set from fine to coarse label levels

References

1. Pawlak, Z. Rough Sets. *Int. J. Comput. Inf. Sci.* **1982**, *11*, 341–356. [\[CrossRef\]](#)
2. Cai, M.; Yan, M.; Wang, P.; Xu, F. Multi-label feature selection based on fuzzy rough sets with metric learning and label enhancement. *Int. J. Approx. Reason.* **2024**, *168*, 109149. [\[CrossRef\]](#)
3. Matusiewicz, Z.; Mroczek, T. Attribute reduction method based on fuzzy relational equations and inequalities. *Int. J. Approx. Reason.* **2025**, *178*, 109355. [\[CrossRef\]](#)
4. Xie, Y. Attribute reduction with pessimistic multigranulation rough sets in relation systems. *Int. J. Approx. Reason.* **2025**, *186*, 109515. [\[CrossRef\]](#)
5. Chen, J.; Zhu, P. Discernibility matrix-based feature selection approaches with fuzzy dominance-based neighborhood rough sets. *Fuzzy Sets Syst.* **2025**, *513*, 109384. [\[CrossRef\]](#)
6. Yuan, K.; Xu, W.; Miao, D. A local rough set method for feature selection by variable precision composite measure. *Appl. Soft Comput.* **2024**, *155*, 111450. [\[CrossRef\]](#)
7. Wang, C.; Wang, C.; An, S.; Ding, W.; Qian, Y. Feature selection and classification based on directed fuzzy rough sets. *IEEE Trans. Syst. Man Cybern. Syst.* **2025**, *55*, 699–711. [\[CrossRef\]](#)

8. Zhang, P.; Zhao, Y.; Hu, L.; Wang, D.; Peng, L.; Li, Z.; Unger, H.; Li, T. I2QD: Unsupervised feature selection via information quality, quantity, and difference degree. *Inf. Process. Manag.* **2025**, *62*, 104173. [\[CrossRef\]](#)
9. Chen, Y.; Cheng, Y.; Luo, B.; Shao, Y.; Zhao, M.; Zhang, Q. A novel attribute reduction algorithm based on granular sequential three-way decision. *Inf. Sci.* **2025**, *694*, 121691. [\[CrossRef\]](#)
10. Zhang, G.; Hu, J.; Yang, J.; Zhang, P. Interactive streaming feature selection based on neighborhood rough sets. *Eng. Appl. Artif. Intell.* **2025**, *139*, 109479. [\[CrossRef\]](#)
11. Xu, W.; Yuan, K.; Li, W.; Ding, W. An emerging fuzzy feature selection method using composite entropy-based uncertainty measure and data distribution. *IEEE Trans. Emerg. Top. Comput. Intell.* **2023**, *7*, 76–88. [\[CrossRef\]](#)
12. Ye, J.; Sun, B.; Zhao, X.; Chu, X. A three-way efficacy evaluation approach with attribute reduction based on weighted temporal fuzzy rough sets. *Inf. Sci.* **2025**, *712*, 122157. [\[CrossRef\]](#)
13. Zhang, P.; Zhang, Q.; Liu, J.; Wang, D.; Ye, H.; Zhang, X.; Li, T. Information fusion and feature selection for multi-source data utilizing Dempster-Shafer evidence theory and K-nearest neighbors. *Inf. Sci.* **2025**, *718*, 122408. [\[CrossRef\]](#)
14. Yuan, K.; Miao, D.; Yao, Y.; Zhang, H.; Zhao, X. Feature selection using zentropy-based uncertainty measure. *IEEE Trans. Fuzzy Syst.* **2024**, *32*, 2246–2260. [\[CrossRef\]](#)
15. Zhang, C.; Lu, Z.; Dai, J. Incremental attribute reduction for dynamic fuzzy decision information systems based on fuzzy knowledge granularity. *Inf. Sci.* **2025**, *689*, 121467. [\[CrossRef\]](#)
16. Thuy, N.N.; Wongthanavas, S. A new approach for reduction of attributes based on stripped quotient sets. *Pattern Recognit.* **2020**, *97*, 106999. [\[CrossRef\]](#)
17. Qian, Y.; Liang, J.; Pedrycz, W.; Dang, C. Positive approximation: An accelerator for attribute reduction in rough set theory. *Artif. Intell.* **2010**, *174*, 597–618. [\[CrossRef\]](#)
18. Ding, W.; Qin, T.; Shen, X.; Ju, H.; Wang, H.; Huang, J.; Li, M. Parallel incremental efficient attribute reduction algorithm based on attribute tree. *Inf. Sci.* **2022**, *610*, 1102–1121. [\[CrossRef\]](#)
19. Li, B.; Wei, Z.; Miao, D.; Zhang, N.; Shen, W.; Gong, C.; Zhang, H.; Sun, L. Improved general attribute reduction algorithms. *Inf. Sci.* **2020**, *536*, 298–316. [\[CrossRef\]](#)
20. Chen, Y.; Liu, K.; Song, J.; Fujita, H.; Yang, X.; Qian, Y. Attribute group for attribute reduction. *Inf. Sci.* **2020**, *535*, 64–80. [\[CrossRef\]](#)
21. Xia, S.; Bai, X.; Wang, G.; Cheng, Y.; Meng, D.; Gao, X.; Zhai, Y.; Giem, E. An efficient and accurate rough set for feature selection, classification, and knowledge representation. *IEEE Trans. Knowl. Data Eng.* **2023**, *35*, 7724–7735. [\[CrossRef\]](#)
22. Wang, G.; Zhang, P.; Wang, D.; Chen, H.; Li, T. Fast attribute reduction via inconsistent equivalence classes for large-scale data. *Int. J. Approx. Reason.* **2023**, *163*, 109039. [\[CrossRef\]](#)
23. Feng, Q.; Miao, D.; Cheng, Y. Hierarchical decision rules mining. *Expert Syst. Appl.* **2010**, *37*, 2081–2091. [\[CrossRef\]](#)
24. Wu, W.Z.; Leung, Y. Theory and applications of granular labelled partitions in multi-scale decision tables. *Inf. Sci.* **2011**, *181*, 3878–3897. [\[CrossRef\]](#)
25. Yang, Y.; Zhang, Q.; Zhao, F.; Cheng, Y.; Xie, Q.; Wang, G. Optimal scale combination selection based on genetic algorithm in generalized multi-scale decision systems for classification. *Inf. Sci.* **2025**, *693*, 121685. [\[CrossRef\]](#)
26. Wang, L.X.; Wu, W.Z.; Xie, Z.H.; Tan, A. Optimal scale combination selection in generalized multi-scale hybrid decision systems. *Inf. Sci.* **2025**, *689*, 121429. [\[CrossRef\]](#)
27. Huang, Z.; Li, J.; Dai, W.; Lin, R. Generalized multi-scale decision tables with multi-scale decision attributes. *Int. J. Approx. Reason.* **2019**, *115*, 194–208. [\[CrossRef\]](#)
28. Li, S.; Yang, J.; Wang, G.; Zhang, Q.; Hu, J. Granularity selection for hierarchical classification based on uncertainty measure. *IEEE Trans. Fuzzy Syst.* **2022**, *30*, 4841–4855. [\[CrossRef\]](#)
29. Qiu, Z.; Hu, M.; Zhao, H. Hierarchical classification based on coarse-to-fine knowledge transfer. *Int. J. Approx. Reason.* **2022**, *149*, 61–69. [\[CrossRef\]](#)
30. Zhao, H.; Wang, P.; Hu, Q.; Zhu, P. Fuzzy rough set based feature selection for large-scale hierarchical classification. *IEEE Trans. Fuzzy Syst.* **2019**, *27*, 1891–1903. [\[CrossRef\]](#)
31. Qiu, Z.; Zhao, H. A fuzzy rough set approach to hierarchical feature selection based on Hausdorff distance. *Appl. Intell.* **2022**, *52*, 11089–11102. [\[CrossRef\]](#)
32. Yang, J.; Qin, X.; Wang, G.; Zhang, Q.; Li, S.; Wu, D. Attribute reduction for hierarchical classification based on improved fuzzy rough set. *Inf. Sci.* **2024**, *677*, 120900. [\[CrossRef\]](#)
33. Lin, Z.; Lin, Y.; Zhang, Y.; Zhao, H. Hierarchical feature selection driven by inter-category relevance and semantic independence. *Pattern Recognit.* **2025**, *161*, 111335. [\[CrossRef\]](#)
34. Yang, X.; Li, Y.; Liu, D.; Li, T. Hierarchical fuzzy rough approximations with three-way multigranularity learning. *IEEE Trans. Fuzzy Syst.* **2022**, *30*, 3486–3500. [\[CrossRef\]](#)
35. Huang, W.; She, Y.; He, X.; Ding, W. Fuzzy rough sets-based incremental feature selection for hierarchical classification. *IEEE Trans. Fuzzy Syst.* **2023**, *31*, 3721–3733. [\[CrossRef\]](#)

36. Wang, Y.; Hu, Q.; Chen, H.; Qian, Y. Uncertainty instructed multi-granularity decision for large-scale hierarchical classification. *Inf. Sci.* **2022**, *586*, 644–661. [\[CrossRef\]](#)
37. Qian, J.; Hong, C.; Yu, Y.; Liu, C.; Miao, D. Generalized multigranulation sequential three-way decision models for hierarchical classification. *Inf. Sci.* **2022**, *616*, 66–87. [\[CrossRef\]](#)
38. Qian, J.; Tang, D.; Yu, Y.; Yang, X.; Gao, S. Hierarchical sequential three-way decision model. *Int. J. Approx. Reason.* **2022**, *140*, 156–172. [\[CrossRef\]](#)
39. Qian, J.; Liu, C.; Miao, D.; Yue, X. Sequential three-way decisions via multi-granularity. *Inf. Sci.* **2020**, *507*, 606–629. [\[CrossRef\]](#)
40. Qian, J.; Zheng, M.; Yu, Y.; Zhou, C.; Miao, D. A dynamic anonymization privacy-preserving model based on hierarchical sequential three-way decisions. *Inf. Sci.* **2025**, *686*, 121316. [\[CrossRef\]](#)
41. Miao, D.; Zhao, Y.; Yao, Y.; Li, H.X.; Xu, F. Relative reducts in consistent and inconsistent decision tables of the Pawlak rough set model. *Inf. Sci.* **2009**, *179*, 4140–4150. [\[CrossRef\]](#)
42. Hu, X.; Cercone, N. Learning in relational databases: A rough set approach. *Comput. Intell.* **1995**, *11*, 323–338. [\[CrossRef\]](#)
43. Zhang, X.; Yao, H.; Lv, Z.; Miao, D. Class-specific information measures and attribute reducts for hierarchy and systematicness. *Inf. Sci.* **2021**, *563*, 196–225. [\[CrossRef\]](#)
44. Zhang, X.; Yao, Y. Tri-level attribute reduction in rough set theory. *Expert Syst. Appl.* **2022**, *190*, 116187. [\[CrossRef\]](#)
45. Wei, W.; Wang, J.; Liang, J.; Mi, X.; Dang, C. Compacted decision tables based attribute reduction. *Knowl.-Based Syst.* **2015**, *86*, 261–277. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.