International Journal of Approximate Reasoning 187 (2025) 109541

Contents lists available at ScienceDirect

International Journal of Approximate Reasoning

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

Recovery degree constrained equiconcept/pseudo-equiconcept reduction in symmetric formal contexts

Junyu Bu [a], Fei Hao [a,*], Huilin Fan [a], Ling Wei [b], Shuanggen Liu [c], Sergei O. Kuznetsov [d]

[a] School of Artificial Intelligence and Computer Science, Shaanxi Normal University, Xi'an, 710119, China
[b] School of Mathematics, Northwest University, Xi'an, 730070, China
[c] School of Cyberspace Security, Xi'an University of Posts and Communications, Xi'an, 710121, China
[d] School of Data Analysis and Artificial Intelligence, National Research University Higher School of Economics, Moscow, Russia

ARTICLE INFO

Dataset link: https://networkrepository.com/index.php

Keywords:
FCA
Symmetric formal context
Equiconcept reduction
Recovery degree
Pseudo-equiconcept reduction

ABSTRACT

In Formal Concept Analysis (FCA), concept reduction serves as an important means of simplification. The application scenarios of concept reduction cover various aspects such as data mining, knowledge discovery, strategic decision-making, and rule learning. For symmetric formal contexts, a specialized class of concept reduction exists that can fully recover all knowledge. However, most existing concept reduction algorithms are designed to recover complete knowledge, which poses limitations in various real-world applications. To this end, this paper proposes a basic recovery degree-constrained equiconcept reduction algorithm, termed *RdER*, enabling knowledge recovery to a specified extent. Additionally, to reduce its running time, an evolutionary algorithm, termed *RdER+*, is further developed. Meanwhile, given the simplicity of pseudo-equiconcepts, we also develop a recovery degree-constrained pseudo-equiconcept reduction algorithm, termed *RdPR*. A large number of experiments have demonstrated that *RdER+* and *RdPR* have significantly reduced the running time while maintaining a relatively low redundancy and ensuring the average recovery degree. Particularly, when the dimension reaches 17, the running speeds of *RdER+* and *RdPR* are, on average, 150 times faster than that of *RdER*. Moreover, as the dimension continues to increase, this advantage in speed will become even more pronounced.

1. Introduction

Formal Concept Analysis (FCA) originated in the 1980s and was proposed by the German mathematician Wille [1]. Its main purpose is to analyze and describe the conceptual structure hidden in data with a formal way, especially the relationships between attributes and objects. FCA identifies and abstracts the hierarchical structure of concepts by constructing a formal context. These concepts are represented by a formal concept lattice [1]. Importantly, FCA has a wide range of application fields, including information systems [2], data mining [3–5], three-way decision [6–10], social network analysis [11–13], and natural language processing [14,15], etc.

Concept reduction [16,17] is a reduction theory in FCA. By eliminating redundant concepts, the results of the analysis become more concise and easier to understand, while also avoiding duplicate information. This allows for a clearer focus on the core structure and patterns within the data. In the case of a symmetric formal context [18], the result of concept reduction exhibits a unique symmetry.

* Corresponding author.
E-mail address: feehao@gmail.com (F. Hao).

https://doi.org/10.1016/j.ijar.2025.109541
Received 2 June 2025; Received in revised form 3 July 2025; Accepted 30 July 2025

Available online 6 August 2025

Specifically, the reduction process results in a set of symmetric concepts. If a core concept exists, it must be symmetric and is referred to as a symmetric concept [18]. In fact, symmetric concepts correspond to equiconcepts [18,11], which are defined as concepts that share equal extent and intent.

Wei et al. [18] proposed concept reduction in a symmetric formal context, which successfully preserves the invariance of binary relations. Symmetric formal contexts have many applications in reality. For example, in memorizing geographical knowledge: taking countries as objects, the relationship between two countries in the same continent is marked as 1, and 0 if they are in different continents. Forming equiconcepts through concept reduction can display the relationships among multiple countries more clearly, thereby reducing the difficulty of memory. Another example is in memorizing biological knowledge: taking species as objects, the relationship between two species in the same family is marked as 1, and 0 if they are different. This symmetric context can also form equiconcepts through concept reduction, helping people find the connections between things.

However, in some practical applications, it is often not necessary to fully preserve all binary relations to achieve the expected goal. To the best of our knowledge, there are no scholars who have conducted in-depth discussions on the flexibility and efficiency issues in these specific situations. The following are the typical application scenarios of recovery degree constrained equiconcept reduction in symmetric formal contexts.

- Scenario 1 (Educational Assessment [19]): If the knowledge that students need to memorize is abstracted into a symmetric formal context, generally speaking, students only need to master a certain extent of knowledge to pass the exam. For example, mastering 70% of the knowledge is sufficient. This reflects the “recovery degree constrained equiconcept reduction”, that is, one doesn’t need to master all knowledge points and can succeed when reaching a certain threshold.
- Scenario 2 (Inference and Evidence in Archaeology [20]): If the connections among the contents of ancient books are abstracted into a symmetric formal context, the restoration of ancient books usually can’t restore them to 100% of their original appearance. On the contrary, when the restoration degree reaches a certain level, the core value of the literature can still be retained.
- Scenario 3 (Ecological restoration [21]): In terms of ecological restoration, if the relationships among organisms in the ecosystem are abstracted into a symmetric formal context, then only a certain degree of key ecological functions need to be restored, and the ecosystem can also maintain its stability.

Obviously, these application fields all indicate that in complex systems, achieving a certain recovery degree threshold can achieve the goal, rather than pursuing perfect restoration.

As a deformation of the equiconcept, the pseudo-equiconcept reduction [22] simplifies the process of concept reduction by removing duplicate relations, making the final result more concise and clear. The pseudo-equiconcepts help refine and optimize concept reduction, making the calculation more efficient and intuitive.

Therefore, the objectives of our research are as follows:

- Objective1: Apply the recovery degree to the equiconcept/pseudo-equiconcept reduction algorithm.
- Objective2: Use as few equiconcepts as possible to recover as many relationships as possible.

Therefore, in order to achieve the above objectives, we have respectively designed a recovery degree constrained equiconcept reduction algorithm (RdER) and a recovery degree constrained pseudo-equiconcept reduction algorithm (RdPR). Besides, due to the overly long running time of RdER, we have designed an ievolutionary algorithm (RdER+) to reduce the time complexity. The main contributions of this paper can be summarized as follows:

- Recovery Degree Constrained Equiconcept Reduction Algorithm (RdER): In a symmetric formal context, this paper proposes and defines “number of reference relationships ($\mathscr{F}$)” and “number of recovery relationships (C)”, and deduces the calculation formula of C. By establishing a systematic relation matrix and analyzing the transformation, the formula can accurately calculate the recovery degree after transformation. Using this formula, we implement a recovery degree constrained equiconcept reduction algorithm. This algorithm optimizes and simplifies the equivalent states of the system while retaining the recovery degree information of the system. This reduction not only makes model calculation more efficient but also can more accurately reflect the behavior of the system when considering perturbations and transformations.
- Recovery Degree Constrained Equiconcept Reduction Evolutionary Algorithm (RdER+): Meanwhile, to address the problem of excessive running time in the RdER, we propose and define “number of recovery equiconcepts ($\mathscr{E}$)” and “number of relationships of derived from recovery equiconcept ($\mathscr{R}$)”. Based on these, an evolutionary algorithm is implemented to reduce the number of selection operations in the algorithm, thereby reducing the time cost.
- Recovery Degree Constrained Pseudo-equiconcept Reduction Algorithm (RdPR): For the pseudo-equiconcept in the symmetric formal context, we have implemented a recovery degree constrained pseudo-equiconcept reduction algorithm. This algorithm has a shorter running time and higher running efficiency, demonstrating the superiority of the pseudo-equiconcept.
- Evaluation: We conducted performance tests of the three algorithms on ten artificially synthesized datasets and three real-world datasets respectively, and provided three evaluation metrics, namely running time, redundancy, and average recovery degree. We hope that the running time and redundancy are as small as possible, while the average recovery degree is as large as possible. Theoretical analysis and experimental results show that each of the three algorithms has its own advantages. Among them, RdER performs well when the dataset is small and the recovery degree is low. When the recovery degree increases, although this algorithm has a longer running time, the redundancy and average recovery degree are 0.5 times and 2 times those of RdER+.

RdER+ is more suitable for situations where the dataset is large and the recovery degree is low. As the recovery degree increases, its running time decreases significantly compared with RdER algorithm. For RdPR, it is more universal, and the redundancy is always 0, but its performance in terms of average recovery degree deteriorates as the dataset size increases.

The rest of this paper is structured as follows: Section 2 introduces the related work. Section 3 outlines the preliminary knowledge and problem formulation. Section 4 presents the proposed RdER and RdER+. Section 5 introduces RdPR. Section 6 shows the experimental details and result analysis. Finally, Section 7 concludes this paper.

2. Related work

As the research background of FCA, formal context has received much attention. The formal context of adding the decision attribute is called the formal decision context [23–25]. The ambiguous context [26,27] and the interval value context [28,29] are composed of uncertain data. Wang et al. [30] defined the coalition formal context and applied it in conflict analysis. Hao et al. [11] proposed a formal context based on the modified adjacency matrix and used it to detect k-cliques in social networks. Both of them have internal reflexive and symmetric relations and can be represented in the symmetric formal context. Thus, the symmetric formal context is of great significance for FCA.

As a hot issue in FCA, the reduction problem mainly has two categories. The first category is attribute reduction, that is, to ensure a certain property of the formal context remains unchanged with a minimal attribute subset. For example, Zhang et al. [31] proposed the attribute reduction theory on the concept lattice, which ensured that under the premise of the invariance of the object set, a minimal attribute subset was found so that the concept lattice determined by it was isomorphic to the concept lattice determined by all attributes. In addition, there are also attribute reductions about granular computing [32], fuzzy object [33] and irreducible elements [34]. The second category is concept reduction, that is, to ensure the relation of the formal context remains unchanged with a minimal concept subset. On this basis, Xie et al. [35] studied the concept characteristics and concept reduction problems of keeping the binary relation unchanged through Boolean matrix operations. Wang et al. [36] proposed a method of attribute-oriented concept coordination set and concept reduction. Ren et al. [37] proposed the three-way approximate concept reduction theory in the incomplete formal context, covering both object and attribute aspects. Li et al. [38] proposed a ternary concept reduction method, which expanded the binary theory.

In the existing concept reduction algorithms, the invariance of the relation is taken as the premise. However, in some real-world fields, the flexibility and efficiency of relation selection urgently need to be studied. Therefore, this paper proposes RdER. This algorithm can effectively recover the information of the symmetric formal context from some equiconcepts to achieve the specified recovery degree. Further improvement of this algorithm is made, and RdER+ is proposed. In addition, considering the compactness and flexibility of the pseudo-equiconcept, RdPR is also designed. The performance of the algorithms is tested on three real datasets, and the evaluation metrics are running time, redundancy, and average recovery degree respectively. The experimental results show that the three algorithms have their own advantages in formal context recovery.

3. Preliminary knowledge and problem formulation

In this section, we first introduce the preliminary knowledge of FCA methodologies, and then describe the problem of how to partially restore the information represented by the symmetric formal context.

3.1. Basic definitions in FCA

FCA is a mathematical tool and method for systematic analysis of concepts in datasets. It represents the hierarchical relationship between concepts by establishing a concept lattice, thereby revealing the structure and patterns in the data. Its input data and structure is provided by the formal context, which consists of a triple composed of an object set, an attribute set, and a binary relationship (object-attribute), representing the relationship between objects and attributes.

Definition 1. (***Formal Context***) [1] Let a formal context be defined as a triple (O, A, I), O is the object set; A is the attribute set; $I \subseteq O \times A$ is the binary relationship between the object set O and the attribute set A. For any object $o \in O$ and attribute $a \in A$: if $(o, a) \in I$, then it is said that object o has attribute a.

Definition 2. (***Relation Matrix*** R) [16] For a formal context (O, A, I), where $O = \{o_1, o_2, \cdots, o_p\}$, $A = \{a_1, a_2, \cdots, a_q\}$, and I is the relationship between objects and attributes, then the relation matrix R is a $p \times q$ Boolean matrix, where:

$$R = [R_{ij}]_{p\times q}, \qquad R_{ij} = \begin{cases} 1, & (o_i, a_j) \in I, \\ 0, & (o_i, a_j) \notin I. \end{cases} \tag{1}$$

The elements of O and A must be arranged in a predefined order. Specifically, the i-th row of R corresponds to the i-th object in the ordered sequence of O, and the j-th column corresponds to the j-th attribute in the ordered sequence of A.

Definition 3. (***Extent and Intent***) [1] Given a formal context (O, A, I), the operators $\uparrow$ and $\downarrow$ applied to subsets of O and A are defined as follows:

$$X^{\uparrow} = \{a \in A \mid \forall o \in X, (o,a) \in I\},$$
$$Y^{\downarrow} = \{o \in O \mid \forall a \in Y, (o,a) \in I\}. \tag{2}$$

If $X^{\uparrow} = Y$ and $Y^{\downarrow} = X$, then (X,Y) is called a formal concept. X is called the intent and Y is called the extent of the formal concept (X,Y). To facilitate the subsequent discussion, let l_i denote the (i+1)-th concept ID.

Definition 4. (***Concept Lattice***) [1] Let $\mathrm{L}(O,A,I)$ denote the set of all concepts of the formal context (O,A,I). Then, the partial order relation of concepts is defined as follows:

$$(X_1,Y_1) \le (X_2,Y_2) \Leftrightarrow X_1 \subseteq X_2 (\Leftrightarrow Y_1 \supseteq Y_2). \tag{3}$$

The partial order $(\mathrm{L}(O,A,I), \le)$ forms an algebraic lattice, which is called the concept lattice.

Definition 5. (***Concept Reduction***) [16] Let $\mathrm{L}(O,A,I)$ denote the concept lattice of the formal context (O,A,I), and $\mathcal{F}$ be any subset of $\mathrm{L}(O,A,I)$. If

$$I = \bigcup_{(X,Y)\in\mathcal{F}} X \times Y, \tag{4}$$

then $\mathcal{F}$ is called a concept consistent set that keeps the binary relation unchanged, referred to as a concept coordination set for short. If for any $(U,V) \in \mathcal{F}$, the following condition holds:

$$I \neq \bigcup_{(X,Y)\in\mathcal{F}\setminus\{(U,V)\}} X \times Y, \tag{5}$$

then $\mathcal{F}$ is called a concept reduct that keeps the binary relation unchanged, referred to as a concept reduction for short.

The calculations of the mentioned concept reduction can be applied to numerous scenarios, such as in supermarket operations, market basket analysis utilizes concept reduction to uncover customer purchase patterns and assist supermarkets in decision-making. The specific steps are as follows:

1) Construct the formal context: Based on customer shopping data, let the customer set be $C = \{C_1, C_2, \cdots, C_n\}$ and the product set be $P = \{P_1, P_2, \cdots, P_m\}$. If customer C_i purchases product P_j, there is a record in the purchase relationship I. These three elements form the formal context (C,P,I). For example, C_1 buys bread, milk, and jam, and C_2 buys bread, eggs, and ham sausages. This information is reflected in I.
2) Generate the concept lattice: By using the formal concept analysis method, the concepts in the generated concept lattice are composed of an extent (the set of customers who purchase certain products) and an intent (the set of products jointly purchased by certain customers). For instance, the extent of a certain concept is $\{C_1, C_3, C_5\}$ who buy bread, milk, and jam, and its intent is $\{\text{bread}, \text{milk}, \text{jam}\}$. As the number of products and customers increases, the concept lattice becomes more complex.
3) Perform concept reduction: Identify important concepts by focusing on high-frequency product-combination concepts, such as the "bread + milk + jam" combination. Remove redundant concepts by eliminating the concepts of niche products that are occasionally purchased alone, ensuring that key purchase-pattern information is not lost. For example, by removing the concept corresponding to toothpicks and retaining the core combination, the main purchase patterns can still be described.
4) Apply the reduction results: Optimize product display by placing high-frequency co-purchased products together. For example, place bread, milk, and jam adjacent to each other to enhance the convenience of shopping and increase sales. Develop promotional strategies by launching preferential packages for high-frequency co-purchased combinations, such as the "bread + milk + jam" package, which can increase the sales of these products, drive the sales of related products, meet customer needs, and enhance customer satisfaction.

3.2. Symmetric formal context

Definition 6. (***Symmetric Formal Context***) [18] A formal context (O,O,I) is called a symmetric formal context, where O represents the set of objects/attributes.

For the better presentation, we denote the above symmetric formal context (O,O,I) as K.

Example 1. Let a symmetric formal context $K_0 = (O,O,I)$, O represents the set of balloons, where $O = \{1,2,3,4,5\}$. The binary relation I is used to describe whether the colors of balloons are the same. The relation matrix R under this symmetric formal context is shown in Table 1. And Table 2 shows the set of all concepts of this context.

Table 1
The relation matrix R.

Object	1	2	3	4	5
1	1	1	1	1	0
2	1	1	1	0	1
3	1	1	1	1	1
4	1	0	1	1	1
5	0	1	1	1	1

Table 2
All formal concepts of K_0.

Concept ID	formal concept
l_0	$(\emptyset, O)$
l_1	$(\{3\}, \{1,2,3,4,5\})$
l_2	$(\{1,3\}, \{1,2,3,4\})$
l_3	$(\{2,3\}, \{1,2,3,5\})$
l_4	$(\{3,4\}, \{1,3,4,5\})$
l_5	$(\{3,5\}, \{2,3,4,5\})$
l_6	$(\{1,2,3\}, \{1,2,3\})$
l_7	$(\{1,3,4\}, \{1,3,4\})$
l_8	$(\{1,3,5\}, \{2,3,4\})$
l_9	$(\{2,3,4\}, \{1,3,5\})$
l_{10}	$(\{2,3,5\}, \{2,3,5\})$
l_{11}	$(\{3,4,5\}, \{3,4,5\})$
l_{12}	$(\{1,2,3,4\}, \{1,3\})$
l_{13}	$(\{1,2,3,5\}, \{2,3\})$
l_{14}	$(\{1,3,4,5\}, \{3,4\})$
l_{15}	$(\{2,3,4,5\}, \{3,5\})$
l_{16}	$(\{1,2,3,4,5\}, \{3\})$
l_{17}	$(O, \emptyset)$

3.3. Equiconcept and pseudo-equiconcept

Definition 7. (***Equiconcept***) [39] Given a symmetric formal context K, for a formal concept (X, Y), if $X = Y$, then concept (X, Y) is called an equiconcept. The set of all equiconcepts $\mathrm{SC}^{(K)}$ is defined as follows:

$$\mathrm{SC}^{(K)} = \{(X, Y) \in \mathrm{L}(K) \mid X = Y\}. \tag{6}$$

Theorem 1. *[18] Given a symmetric formal context (O, O, I), $\mathrm{SC}^{(K)}$ corresponds to a special concept reduct under this context.*

Proof. Detailed proofs can be found in the reference [18]. □

Definition 8. (***Pseudo-equiconcept***) [22] Let (O, O, I) be a symmetric formal context and $X \subseteq O$. If $X \subseteq X^{\uparrow}$ and $|X| \geq 1$, (X, X) is called a pseudo-equiconcept. Here, X is both the extent and the intent of the pseudo-equiconcept. The set of all pseudo-equiconcepts is denoted by $\mathrm{T}(O, O, I)$.

Definition 9. (***Pseudo-equiconcept reduction***) [22] Consider a symmetric formal context (O, O, I) and $\mathcal{T} \subseteq \mathrm{T}(O, O, I)$. If the following equation holds:

$$I = \bigcup_{(X,X)\in\mathcal{T}} X \times X, \tag{7}$$

then $\mathcal{T}$ is referred to as a pseudo-equiconcept consistent set that preserves the binary relation. Furthermore, if for any $(Y, Y) \in \mathcal{T}$, the following condition holds:

$$I \neq \bigcup_{(X,X)\in\mathcal{T}\setminus\{(Y,Y)\}} X \times X, \tag{8}$$

then $\mathcal{T}$ is referred to as a pseudo-equiconcept reduct that preserves binary relation of the context.

Example 2. (Continued from Example 1) Under the symmetric formal context K_0, the set of equiconcepts obtained by the reduction of the equiconcept is shown in Table 3, and one of the pseudo-equiconcept reduct is shown in Table 4.

Table 3
All equiconcepts of K_0.

ID	Equiconcept
l_6	$(\{1,2,3\},\{1,2,3\})$
l_7	$(\{1,3,4\},\{1,3,4\})$
l_{10}	$(\{2,3,5\},\{2,3,5\})$
l_{11}	$(\{3,4,5\},\{3,4,5\})$

Table 4
One of the pseudo-equiconcept reduct of K_0.

ID	Pseudo-equiconcept
t_0	$(\{1,2\},\{1,2\})$
t_1	$(\{1,3,4\},\{1,3,4\})$
t_2	$(\{2,3,5\},\{2,3,5\})$
t_3	$(\{4,5\},\{4,5\})$

3.4. Problem statement and solution framework

In this section, we formulate the problem to be addressed in this paper, i.e., how to partially restore the information represented by the symmetric formal context in the form of equiconcepts and pseudo-equiconcepts respectively.

Input: A set of extents of equiconcepts that can restore all binary relations under the symmetric formal context (O,O,I).

Output1 (equiconcept reduction): A subset of the input set that can restore at least a corresponding proportion of binary relations.

Output2 (pseudo-equiconcept reduction): A set of pseudo-equiconcept sets that can restore at least a corresponding proportion of binary relations.

Problem Statement: The above two problems can be considered and solved by the following three steps:

- RQ1: Given a set of extents of equiconcepts, how can we restore the binary relations therein?
- RQ2: Given a recovery degree, how can we determine which binary relations are needed?
- RQ3: According to the obtained binary relations, how can they be output? Is it output in the form of equiconcepts or as a set of binary relations?

Solution Framework: In the symmetric formal context, this paper adopts a backward deduction thought to determine the entire implementation steps of the algorithm. First, we design a function to fully restore all binary relations in the input. Then, according to the derived recovery degree formula, we calculate the number N of binary relations required when a given recovery degree is satisfied. Finally, according to the size of N, when the required output is an equiconcept, we find a subset that meets the conditions in the original input for output; when the required output is a pseudo-equiconcept, we directly output randomly selected binary relations, which is the pseudo-equiconcept.

4. Recovery degree constrained equiconcept reduction algorithm

This section studies recovery degree constrained equiconcept reduction on a symmetric formal context, so that the obtained set of equiconcepts can achieve a higher recovery degree with a smaller number of equiconcepts under the condition of a given recovery degree.

4.1. Definition of recovery degree and measurement of the number of recovery relationships

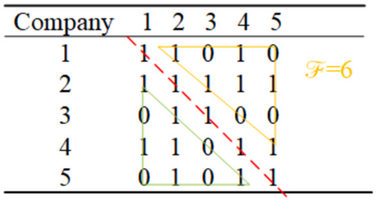
Definition 10. (M_{f} ***and*** M_{c}) Given a symmetric formal context (O,O,I), its relation matrix is denoted by $[R_{ij}]_{|O|\times|O|}$. After applying the recovery degree to (O,O,I), the relation matrix of the resulting symmetric formal context is denoted by $[Q_{ij}]_{|O|\times|O|}$. The reference matrix is defined as follows:

$$M_{\mathrm{f}} = [R_{ij}]_{|O|\times|O|}. \tag{9}$$

The recovery matrix is defined as follows:

$$M_{\mathrm{c}} = [Q_{ij}]_{|O|\times|O|}, \tag{10}$$

where if $R_{ij}=0$ then $Q_{ij}=0$.



Company	1	2	3	4	5
1	1	1	0	1	0
2	1	1	1	1	1
3	0	1	1	0	0
4	1	1	0	1	1
5	0	1	0	1	1

Fig. 1. The Definition of $\mathscr{F}$. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

Table 5
The relation matrix R under K_1.

Company	1	2	3	4	5
1	1	1	0	1	0
2	1	1	1	1	1
3	0	1	1	0	0
4	1	1	0	1	1
5	0	1	0	1	1

Definition 11. The recovery degree under (O,O,I) $\mathscr{D}$ can be calculated by:

$$\mathscr{D}=\frac{N_{\mathrm{c}}}{N_{\mathrm{f}}}, \tag{11}$$

where $N_{\mathrm{f}}=\sum_{i=1}^{|O|}\sum_{j=1}^{|O|}R_{ij}$ and $N_{\mathrm{c}}=\sum_{i=1}^{|O|}\sum_{j=1}^{|O|}Q_{ij}$.

Definition 12. The number of reference relationships under (O,O,I) $\mathscr{F}$ is defined as follows:

$$\mathscr{F}=\frac{N_{\mathrm{f}}-|O|}{2}\quad (0\le\mathscr{F}\le\frac{|O|\times(|O|-1)}{2}). \tag{12}$$

The definition of $\mathscr{F}$ is more vividly illustrated in Fig. 1.

Definition 13. The number of recovery relationships under (O,O,I) C is defined as follows:

$$C=\frac{N_{\mathrm{c}}-|O|}{2}\quad (0\le C\le\mathscr{F}). \tag{13}$$

Definition 14. The number of recovery equiconcepts under K $\mathscr{E}$ is defined as follows:

$$\mathscr{E}=\lceil\mathscr{D}\times N\rceil, \tag{14}$$

where N represents the number of equiconcepts of K.

Definition 15. The number of relationships derived from recovery equiconcepts under K $\mathscr{R}$ is defined as follows:

$$\mathscr{R}=\sum_{i=0}^{\mathscr{E}-1}\binom{k_i}{2}, \tag{15}$$

where k_i $(i\in\{0,1,...,\mathscr{E}-1\})$ represents the number of elements in the extent of the equiconcept l_i $(i\in\{0,1,...,\mathscr{E}-1\})$.

Example 3. Table 5 shows a symmetric formal context $K_1=(O,O,I)$ regarding cooperative relationships. O represents the set of companies participating in the research. In this scenario, there are a total of five companies participating in the research, denoted by $O=\{1,2,3,4,5\}$. The binary relationship I describes whether there is a cooperative relationship between companies. Therefore, the relation matrix R under this symmetric formal context is shown in Table 5.

At this time, let R be the reference matrix, then $|O|=5, N_{\mathrm{f}}=17$, and $\mathscr{F}=(17-5)/2=6$. The corresponding binary relations are: $(1,2)$, $(1,4)$, $(2,3)$, $(2,4)$, $(2,5)$, $(4,5)$. One of the recovery matrices M_{c} is shown in Table 6.

At this time, $|O|=5, N_c=13, C=(13-5)/2=4$. The corresponding binary relations are: $(1,4)$, $(2,3)$, $(2,5)$, $(4,5)$, and $\mathscr{D}=13/17\approx 0.765$.

Example 4. Table 7 illustrates a set of equiconcepts under a symmetric formal context $K_2=(O,O,I)$, where $O=\{1,2,3,4,5,6\}$. Let $\mathscr{D}=0.6$. Then the calculations of $\mathscr{E}$ and $\mathscr{R}$ are as follows.

Table 6
One of M_c with M_f as a reference under K_1.

Company	1	2	3	4	5
1	1	0	0	1	0
2	0	1	1	0	1
3	0	1	1	0	0
4	1	0	0	1	1
5	0	1	0	1	1

Table 7
The set of equiconcepts of K_2.

Concept ID	Equiconcept
l_0	$(\{1,2,3,6\},\{1,2,3,6\})$
l_1	$(\{1,2,4\},\{1,2,4\})$
l_2	$(\{4,5\},\{4,5\})$

Table 8
The set of equiconcepts of M_f.

ID	Equiconcept
l_0	$(\{1,2,4\},\{1,2,4\})$
l_1	$(\{2,4,5\},\{2,4,5\})$
l_2	$(\{2,3\},\{2,3\})$

According to Table 4, it can be known that $n=3$. Then $\mathscr{E}=\lceil 0.6\times 3\rceil=2$. If l_0 and l_1 are selected, then $k_0=4, k_1=3$. $\mathscr{R}=\binom{k_0}{2}+\binom{k_1}{2}=4\times 3/2+3\times 2/2=9$.

Based on the $|O|$, Equation (16) reveals the correlations among $\mathscr{F}$, $\mathscr{D}$ and C:

$$\mathscr{D}=\frac{2\times C+|O|}{2\times\mathscr{F}+|O|}\quad(0\le\mathscr{D}\le 1). \tag{16}$$

After transformation and deformation by Equation (16), the calculation formula for the number of recovery relationships is obtained as follows:

$$C=\lceil\mathscr{D}\times\mathscr{F}-\frac{1-\mathscr{D}}{2}\times|O|\rceil\quad(0\le C\le\mathscr{F}). \tag{17}$$

Theorem 2. *When $\mathscr{D}=0$, $C=0$; when $\mathscr{D}=1$, $C=\mathscr{F}$.*

Proof. When $\mathscr{D}=0$, the Equation (17) can be simplified as:

$$C=\lceil 0\times\mathscr{F}-\frac{1-0}{2}\times|O|\rceil=\lceil-\frac{1}{2}\times|O|\rceil\le 0, \tag{18}$$

since $C\ge 0$, so $C=0$.

When $\mathscr{D}=1$, the Equation (17) can be simplified as:

$$C=\lceil 1\times\mathscr{F}-\frac{1-1}{2}\times|O|\rceil=\mathscr{F}.\quad\square \tag{19}$$

4.2. Design ideas of the algorithm

The design flow of recovery degree constrained equiconcept reduction algorithm is shown in Fig. 2.

Input: It specifies $|O|$, $\mathscr{D}$, and the equiconcept of M_f. Since the equiconcept reduction we do is in the context of symmetric form, and the extent and intent of equiconcepts are equal. Therefore, when designing the input, only the extent of the reference matrix needs to be input by the user. At the same time, it sorts the input from large to small according to length to obtain an ordered input.

Example 5. (Continued from Example 3) The set of equiconcepts in the symmetric formal context K_1 in Example 3 is shown in Table 8.

When the user inputs the information of M_f, only the extent of the symmetric formal concept needs to be input: $\{1,2,4\}$, $\{2,3\}$, $\{2,4,5\}$. The program can calculate the $\mathscr{F}$ of M_f according to the input extent.

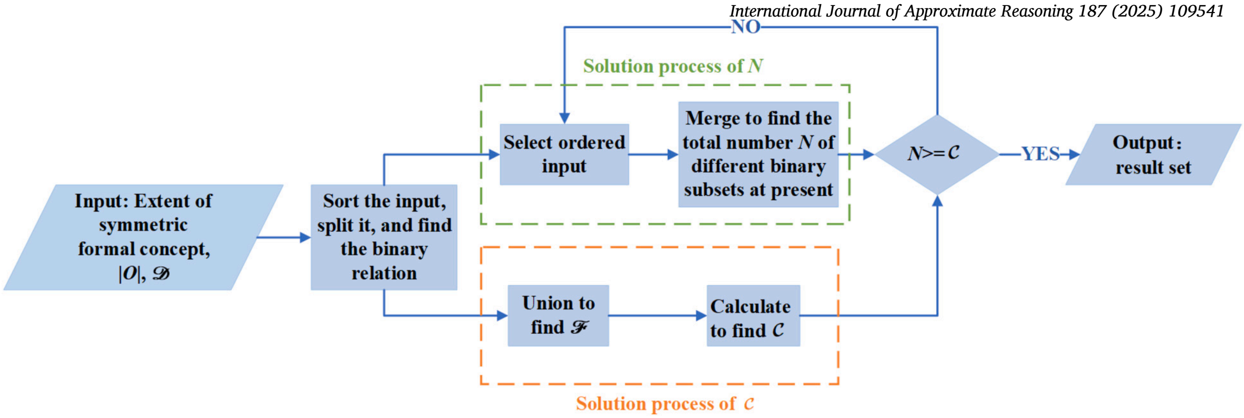


Fig. 2. The design flow of RdER.

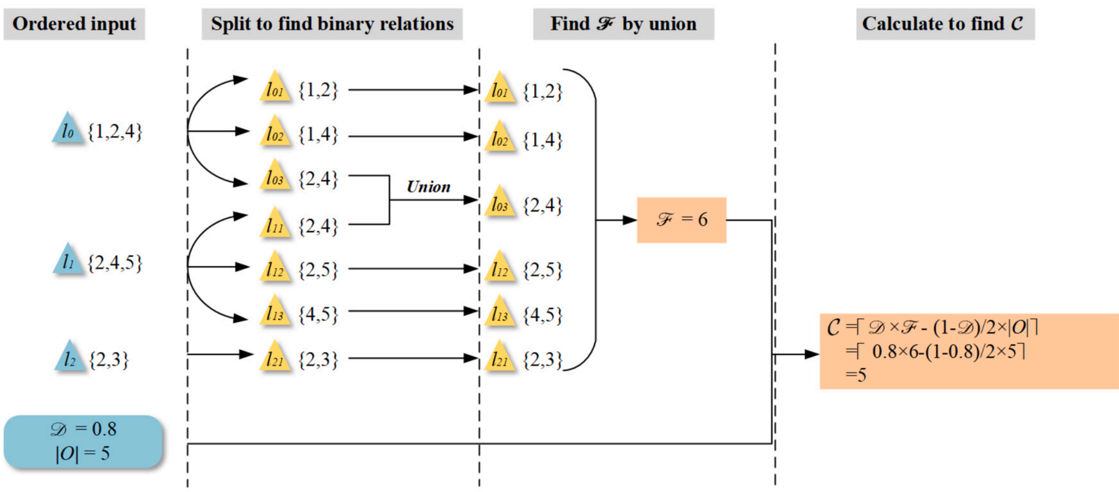


Fig. 3. Calculation process of $\mathscr{F}$ and C.

Greedy strategy: Since a longer extent splits to obtain more binary relations, a higher recovery degree can be achieved with a smaller number of extents. Therefore, we need to sort the input, and preferentially select the longest extent in the current sequence as the result of the current step and add it to the result set.

Binary relation splitting: Binary splitting is performed on the extent set of the equation concept obtained from the input. Combine the elements in the extent in pairs to form subsets of the extent. If there are n elements in the extent, then there will be $\binom{n}{2}$ subsets.

Finding the $\mathscr{F}$ by union: It takes the union operation on all the binary subsets obtained by splitting. Remove duplicate binary relations in the obtained result. At this time, the number of binary subsets is $\mathscr{F}$.

Calculating the C: It substitutes $\mathscr{D}$, $|O|$, and $\mathscr{F}$ obtained by union into Equation (17) to calculate the size of C.

Example 6. (Continued from Example 3) Taking the extents of the set of equation concepts in the symmetric formal context K_1 in Example 3 as input. After sorting the input from long to short, we could obtain $\{1,2,4\}, \{2,4,5\}, \{2,3\}$. Let $\mathscr{D} = 0.8$ and $|O| = 5$. Then the calculation process of $\mathscr{F}$ and C is shown in Fig. 3.

First, we split $\{1,2,4\}$, $\{2,4,5\}$, and $\{2,3\}$ into sets of binary relations respectively. Then merge all the sets of binary relations. Since two sets of $\{2,4\}$ are obtained after splitting, remove the repeated parts after merging and only leave one $\{2,4\}$. Finally, the number of all non-repeated sets of binary relations is $\mathscr{F}$. Further, the C can be calculated according to Equation (17).

Merge judgment: According to the order of the ordered input, the longest extent is firstly added to the result set. We then merge the result set and find the number N of non-repetitive binary subsets in the result set at this time. If $N \geq C$, the result set is output. Otherwise, we merge every two extents in the ordered input and calculate the number N of non-repetitive binary subsets in the result set after each merge. If $N \geq C$, the result set is output at this time. Otherwise, we merge every three extents in the ordered input and calculate the number N of non-repetitive binary subsets in the result set after each merge. Similarly, we may merge every four, merge every five, and so on until $N \geq C$, and then output the result set at this time.

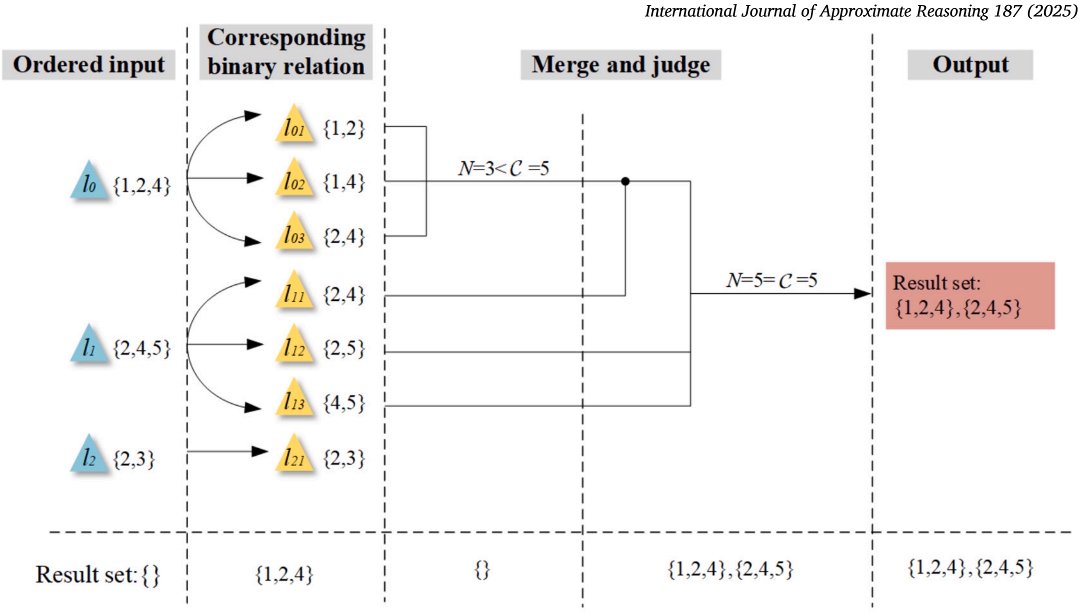


Fig. 4. The process of selective output of results.

Output: To ensure that the output result is still the extent of the symmetric equiconcept, the output result set is designed as the smallest subset of the input set that satisfies the current $\mathscr{D}$, which is called the recovery equiconcept.

Example 7. (Continued from Example 6) After calculating $\mathcal{C}$ and $\mathscr{F}$, the selection and output process of the result is shown in Fig. 4.

We sort $\{1,2,4\}$, $\{2,4,5\}$, and $\{2,3\}$ by length. First, the relation judgment is performed on the longest extent. The result set is initially empty. Then, $\{1,2,4\}$ is added into the result set. Since the number of relations N of $\{1,2,4\}$ is 3, which is less than $\mathcal{C}$, remove $\{1,2,4\}$ from the result set. Then perform pairwise merging. Add $\{1,2,4\}$ and $\{2,4,5\}$ to the result set. Since the number of relations N obtained after merging and removing duplicates of $\{1,2,4\}$ and $\{2,4,5\}$ is 5, which is equal to $\mathcal{C}$, output the result set at this time: $\{1,2,4\}$, $\{2,4,5\}$.

4.3. Algorithm description

The pseudocode of the recovery degree constrained equiconcept reduction algorithm (RdER) designed according to the above design idea is shown as follows:

Algorithm 1 Recovery degree constrained equiconcept reduction algorithm (RdER).

Require:
The extent $l[n]$ of the equiconcept of a symmetric formal context (O, O, I), the number n of extents, the recovery degree R, and the order N

Ensure:
The subset of the input set that satisfies a given recovery degree

$\vartheta \leftarrow \emptyset$, $\Upsilon \leftarrow 0$, $n_0 \leftarrow 0$, $p \leftarrow \emptyset$, $r \leftarrow \emptyset$
for $i \leftarrow 0$ to $n-1$ **do**
  for each $l_{[i][j]}$ in $l_{[i]}$ and $|l_{[i][j]}| = 2$ **do**
    $\vartheta \leftarrow |\vartheta \cup l_{[i][j]}|$
  end for
end for
$\Upsilon \leftarrow \lceil R \times |\vartheta| - (1-R) \times N/2 \rceil$
for $k \leftarrow 2$ to n **do**
  for each p in $l_{[n]}$ and $|p| = k$ **do**
    for each o in p and $|o| = 2$ **do**
      $r \leftarrow r \cup o$
      $n_0 \leftarrow |r|$
      if $n_0 \geq \Upsilon$ **then**
        return p
      end if
    end for
  end for
end for

In lines 1—6 of the code: for each extent, calculating all possible subsets requires traversing all possible binary combinations. Its time complexity is $O(m^2)$, where m is the number of elements in the extent. Suppose the size of each set is roughly the same as m. Since this operation needs to be performed for each extent, if there are n sets in the extent, then the overall time complexity is $O(n \times m^2)$.

Line 7 of the code is used to calculate the C. Lines 8—18 of the code are used for the selection of subsets of the input set.

Generate all combinations pair with length k. The number of combinations is $\binom{n}{k}$, so the time complexity is $\binom{n}{k}$. For each combination, its binary subsets will be merged. This involves merging j subarrays. The time complexity of the merge operation for each subarray is $O(a)$, where a is the total length of the subarray. The total time complexity of the merge operation is $O(j \times a)$. The time complexity of the deduplication operation depends on the number of subarrays and their total length. In the worst case, the complexity of deduplication is $O(j \times a \times \log(j \times a))$.

Therefore, the total time complexity of this algorithm is $O(n \times m^2 + j \times a \times \log(j \times a))$.

4.4. Performance evaluation and improvement of the algorithm

Experiments have found that when the order of the relation matrix gradually increases, the performance of the above algorithm is poor and it is only applicable to small-scale calculations. When the order increases to 20, the running time of this algorithm is too long to obtain accurate results. Therefore, the algorithm is improved and optimized, hoping that it can be applied to large-scale calculations. The optimization idea is as follows:

Since the result set of the above algorithm starts from an empty set and then sequentially adds ordered inputs. For each extent, an addition needs to be performed, the results are merged and deduplicated, and then the size is judged. When the input scale increases, the above process takes too long, resulting in poor performance of the algorithm. When the recovery degree is large, more binary relations need to be recovered. If the result set starts to add extents from an empty set, it will lead to excessive repetition steps and affect the efficiency of the algorithm. Therefore, considering from the perspective of improving the starting point, that is, reducing the running time from the perspective of taking a subset of an input set as the initial value for the result set. Theoretical derivation is as follows:

Suppose l_i $(i \in \{0, 1, ..., n-1\})$ is used as an ordered input, representing that there are n extents of equiconcepts from long to short. Their lengths are correspondingly represented by k_i $(i \in \{0, 1, ..., n-1\})$, and k_i is any positive integer less than or equal to $|O|$. Then the number of corresponding binary relations is $\binom{k_i}{2}$ $(i \in \{0, 1, ..., n-1\})$.

Step 1: Calculate C

$\mathscr{F}$ can be calculated and expressed by the following formula:

$$\mathscr{F} = \sum_{i=0}^{n-1} \binom{k_i}{2} - Q_1, \tag{20}$$

where Q_1 represents the number of repetitions among the binary relations obtained by splitting all inputs. Let recovery degree be $\mathscr{D}$. Then substituting formula (15) into formula (10) gives the expression formula for C as follows:

$$C = \lceil \mathscr{D} \times (\sum_{i=0}^{n-1} \binom{k_i}{2} - Q_1) - \frac{1-\mathscr{D}}{2} \times |O| \rceil, \tag{21}$$

if the result set starts from an empty set, at least one selection needs to be made, that is $k_0 = C$; at most, C selections need to be made, that is $k_i = 1$ $(i \in \{0, 1, ..., n-1\})$.

Step 2: Calculate $\mathscr{R}$

According to the assumed conditions, the calculation formula for $\mathscr{E}$ is as follows:

$$\mathscr{E} = \lceil \mathscr{D} \times n \rceil. \tag{22}$$

Suppose that after sorting the inputs from long to short and sequentially selecting the binary relations obtained by splitting $\mathscr{E}$ equiconcepts, the number of repetitions is Q_2. Then $\mathscr{R}$ can be calculated and expressed by the following formula:

$$\mathscr{R} = \sum_{i=0}^{\lceil \mathscr{D} \times n \rceil - 1} \binom{k_i}{2} - Q_2. \tag{23}$$

Step 3: Calculate the difference between C and $\mathscr{R}$

$$C - \mathscr{R} = \lceil \mathscr{D} \times n \rceil - (\sum_{i=0}^{\lceil \mathscr{D} \times n \rceil - 1} \binom{k_i}{2} - Q_2), \tag{24}$$

since $\mathscr{R} \geq 0$, $C - \mathscr{R} < C$. If there are already $\mathscr{E}$ extents in the result set, at least one selection needs to be made, that is, $k_{\mathscr{E}} = C - \mathscr{R}$; at most $C - \mathscr{R}$ selections need to be made, that is, $k_i = 1 (i \in \{0, 1, ..., n-1\})$.

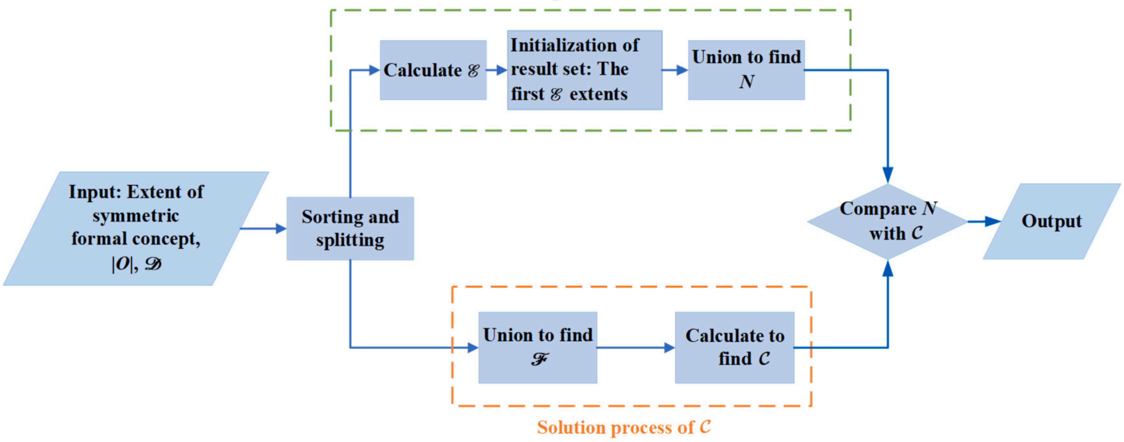


Fig. 5. The design flow of RdER+.

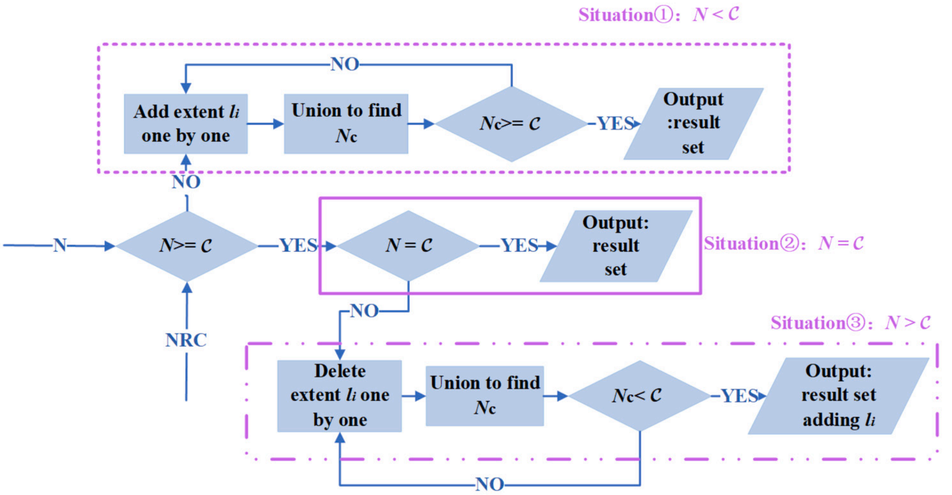


Fig. 6. The flow of "Compare N with C" of RdER+.

Result output:

Let the initial result set contain the first $\mathscr{E}$ extents of the ordered input and calculate the number N of binary relations in the initial result set.

If $N < C$, then add l_i $(i \in \{\mathscr{E}, \mathscr{E}+1, \dots, n-1\})$ to the result set in turn, and calculate the size of the number N_c of binary relations of the result set at this time until $N_c \geq C$, and output the result set.

If $N = C$, then directly output the initial result set.

If $N > C$, then delete l_i $(i \in \{\mathscr{E}-1, \mathscr{E}-2, \dots, 0\})$ from the result set in turn, and calculate the size of the number N_c of binary relations of the result set at this time until $N_c < C$, and output the result set when l_i is not deleted.

The design flow of the evolutionary algorithm is shown in Fig. 5 and Fig. 6.

Example 8. (Continued from Example 6) The calculation process and result of C of the evolutionary algorithm are the same as in Example 6, $C = 5$. Since the number of inputs is 3 and $\mathscr{D} = 0.8$, $\mathscr{E} = \lceil 3 \times 0.8 \rceil = 3$. Therefore, the initial result set is the first three extents of sequentially selected ordered inputs. Then calculate the number N of different binary relations in the result set at this time, and compare N with C. The selection and output process of the result is shown in Fig. 7.

Since $\mathscr{E} = 3$, we first select three extent from the input and add them to the result set. By calculating that the number of non-repeated relations $N_c = 6$ at this time is greater than C, thus we remove $\{2, 3\}$ from the end of the result set. Then the above calculation process is executed again. At this time, $N_c = 5$, which is equal to C, we then remove $\{2, 4, 5\}$ from the end of the result set. Next, by calculating that at this time $N_c = 3$, which is less than C, we output the result set from the previous round while $\{2, 4, 5\}$ was not deleted.

The pseudocode of the main improved part of the evolutionary algorithm (RdER+) is shown in Algorithm 2.

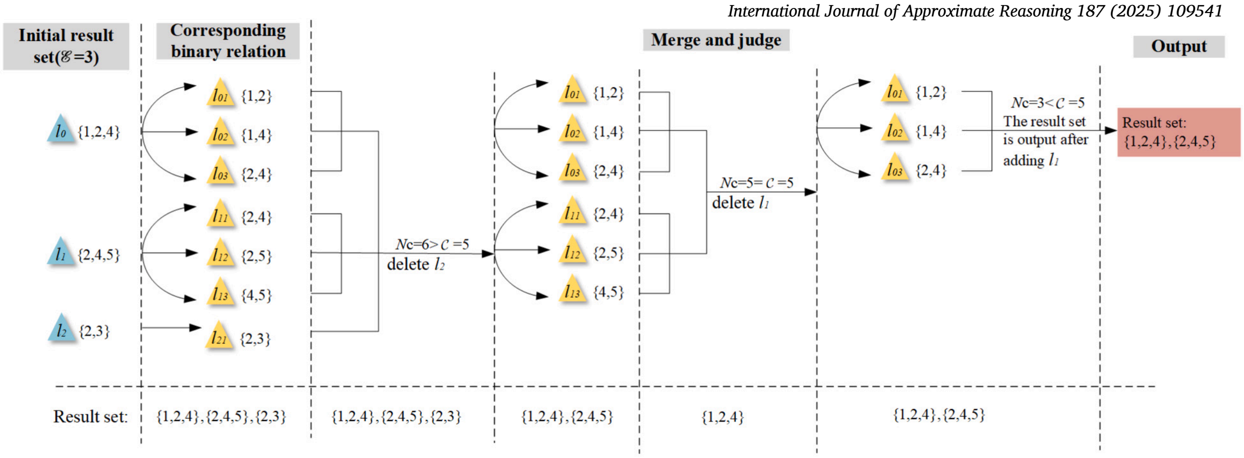


Fig. 7. The process of selective output of the results of the evolutionary algorithm.

Algorithm 2 Recovery degree constrained equiconcept reduction evolutionary algorithm (RdER+).

Require:
The extent $l[n]$ of the equiconcept of a symmetric formal context (O, O, I), the number n of extents, the recovery degree R, and the order N
Ensure:
The subset of the input set that satisfies a given recovery degree

1: Υ, $SC_1 \leftarrow \emptyset$, $SC_2 \leftarrow \emptyset$, $r_1 \leftarrow \emptyset$, $r_2 \leftarrow \emptyset$, $n_1 \leftarrow 0$, $n_2 \leftarrow 0$
2: $\chi \leftarrow \lceil n \times R \rceil$
3: **for** $i \leftarrow 0$ to χ **do**
4: $\quad SC_1 \leftarrow SC_1 \cup l_{[i]}$
5: **end for**
6: $SC_2 \leftarrow l_{[n]} \setminus SC_1$
7: **for** each o_1 in SC_1 and $|o_1| = 2$ **do**
8: $\quad r_1 \leftarrow r_1 \cup o_1$
9: **end for**
10: $n_1 \leftarrow |r_1|$
11: **if** $n_1 = \Upsilon$ **then**
12: $\quad$ **return** SC_1
13: **else if** $n_1 < \Upsilon$ **then**
14: $\quad$ **for** $j \leftarrow 0$ to $|SC_2| - 1$ **do**
15: $\quad\quad SC_1 \leftarrow SC_1 \cup SC_{2[j]}$
16: $\quad\quad$ **for** each o_2 in $SC_{2[j]}$ and $|o_2| = 2$ **do**
17: $\quad\quad\quad r_2 \leftarrow r_1 \cup o_2$
18: $\quad\quad\quad n_2 \leftarrow |r_2|$
19: $\quad\quad\quad$ **if** $n_2 \geq \Upsilon$ **then**
20: $\quad\quad\quad\quad$ **return** SC_1
21: $\quad\quad\quad$ **end if**
22: $\quad\quad$ **end for**
23: $\quad$ **end for**
24: **else**
25: $\quad$ **for** $j \leftarrow |SC_1| - 1$ to 0 **do**
26: $\quad\quad SC_1 \leftarrow SC_1 \setminus SC_{1[j]}$
27: $\quad\quad$ **for** each o_1 in $SC_{1[j]}$ **do**
28: $\quad\quad\quad r_2 \leftarrow r_1 \setminus o_1$
29: $\quad\quad\quad n_2 \leftarrow |r_2|$
30: $\quad\quad\quad$ **if** $n_2 < \Upsilon$ **then**
31: $\quad\quad\quad\quad$ **return** $SC_1 \cup SC_{1[j]}$
32: $\quad\quad\quad$ **end if**
33: $\quad\quad$ **end for**
34: $\quad$ **end for**
35: **end if**

Lines 1 - 6 of the code make a preliminary selection of the input according to the $\mathscr{E}$, lines 7 - 10 calculate the $\mathscr{R}$, and lines 11 - 35 conduct the selection and processing of the result set.

Since the process of calculating all binary subsets of extents is the same as that of RdER, the time complexity of this part is $O(n \times m^2)$.

The time complexity of dividing the input set and calculating its size is $O(k \times l)$, where k and l are the lengths of two lists respectively.

The overall time complexity is dominated by these two parts. Therefore, the total time complexity is $O(n \times m^2 + k \times l)$.

5. Recovery degree constrained pseudo-equiconcept reduction algorithm

This section studies recovery degree constrained pseudo-equiconcept reduction on a symmetric formal context, so that the obtained set of pseudo-equiconcepts can have a lower redundancy under the condition of the given recovery degree.

5.1. Design ideas and steps of the algorithm

The design flow of the recovery degree constrained pseudo-equiconcept reduction algorithm is shown in Fig. 8.

Input: Similar to RdER, it includes $\mathscr{D}$, $|O|$, and the extent of the equiconcept of reference matrix M_f.

Split to find binary relations: Just like RdER, the obtained extent is split into binary subsets to obtain binary relation subsets.

Union to find $\mathscr{F}$ and calculation to find C: Similar to RdER, all binary relation subsets are merged and duplicates are removed. The number of remaining non-repetitive binary relation subsets is the size of $\mathscr{F}$. Then, calculate the size of C according to Equation (17).

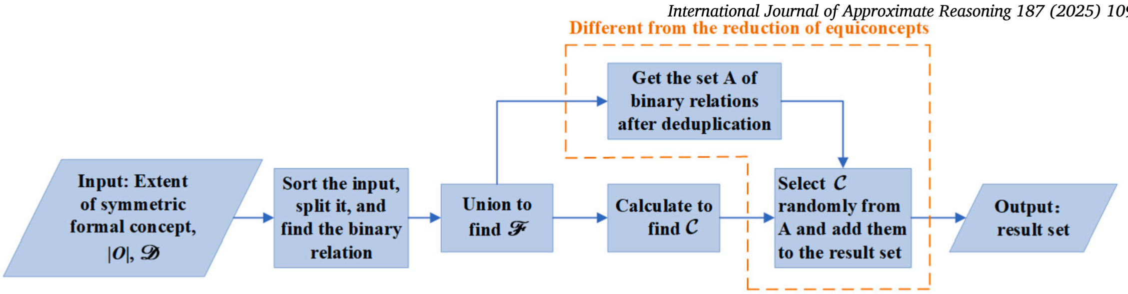


Fig. 8. Flow chart of RdPR.

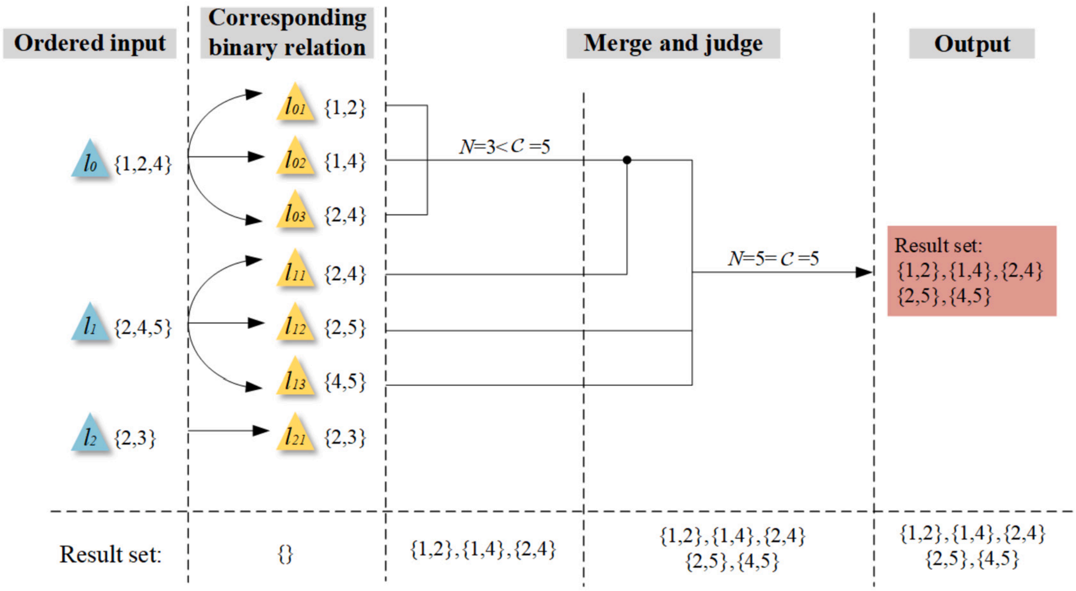


Fig. 9. The process of selective output of results.

Output: Since the binary relation set obtained in the splitting process belongs to the pseudo-equiconcept, therefore, randomly selecting C non-repetitive binary relation sets is the final output result.

Example 9. (Continued from Example 3) Taking the extent of the set of equiconcepts in the symmetric formal context K_1 in Example 3 as input, and after sorting the input from long to short, we get $\{1,2,4\}, \{2,4,5\}, \{2,3\}$. Let $\mathscr{D} = 0.8$ and $|O| = 5$. The calculation process and result of C are the same as in Example 5, and $C = 5$. Then, the selection and output process of the result of the recovery degree constrained pseudo-equiconcept reduction is shown in Fig. 9.

Clearly, the result set starts from empty set. After sorting the input in descending order of length, all of them are split into sets of binary relations. We add the binary relation groups obtained by splitting the longest extent $\{1,2,4\}$ to the result set. After removing duplicates, we calculate the length of the result set $N = 3$, which is less than $C = 5$. After that, we add the binary relation groups obtained by splitting $\{2,4,5\}$ to the result set. After removing duplicates, we calculate the length of the result set $N = 5$, which is equal to $C = 5$. So output the result set at this time: $\{1,2\}, \{1,4\}, \{2,4\}, \{2,5\}, \{4,5\}$.

5.2. Algorithm description

The pseudocode of the recovery degree constrained pseudo-equiconcept reduction algorithm (RdPR) developed according to the above design idea is shown in Algorithm 3.

In lines 1—6 of the code: for each extent, calculating all possible subsets requires traversing all possible binary combinations. Line 7 of the code is used to calculate the C. Lines 8—9 of the code perform the selection and output of the binary relation.

Since the process of calculating all binary subsets of extents is the same as that of Algorithm 1, the time complexity of this part is $O(n \times m^2)$.

Randomly selecting elements from the result has a time complexity of $O(k)$, where k is the number of selections. In the worst case, k is equal to the size of the result.

Algorithm 3 Recovery degree constrained pseudo-equiconcept reduction algorithm (RdPR).

Require:
The extent $l[n]$ of the equiconcept of a symmetric formal context (O, O, I), the number n of extents, the recovery degree R, and the order N
Ensure:
The set of pseudo-equiconcepts satisfying a given recovery

$\vartheta \leftarrow \emptyset$, $\Upsilon \leftarrow 0$, $r \leftarrow \emptyset$
for $i \leftarrow 0$ to $n-1$ **do**
  for each $l_{[i][j]}$ in $l_{[i]}$ and $|l_{[i][j]}| = 2$ **do**
    $\vartheta \leftarrow |\vartheta \cup l_{[i][j]}|$
  end for
end for
$\Upsilon \leftarrow \lceil R \times |\vartheta| - (1-R) \times N/2 \rceil$
$r \leftarrow$ randomly choose Υ of χ (χ in ϑ and $|\chi| = 2$)
return r

Table 9
Dataset Statistics.

Dataset	Type	Nodes	Edges	Average degree
Jazz	Miscellaneous Networks	198	2.7k	27
Celegans-dir	Biological Networks	453	4.6k	20
CSphd	Collaboration Networks	1.9k	1.7k	1

Combining the above parts, the overall time complexity of the algorithm is: $O(n \times m^2 + k)$, where n is the number of sets, m is the maximum size of each set, and k is the number of selections calculated according to the $\mathscr{D}$.

6. Experiments

In this section, we comprehensively evaluate the performance of the above proposed three algorithms. Experiments 1 and 2 are conducted using artificially synthesized datasets. The purpose of Experiment 1 is to comparatively evaluate the influence of matrix dimension and recovery degree on the time performance of the three algorithms. Experiment 2 evaluates the effectiveness of the three algorithms under different recovery degrees based on redundancy and average recovery degree respectively. Experiment 3 uses real datasets to evaluate the effectiveness of RdER+ and RdPR. All algorithms are implemented in Python language and run on a 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz 2.42 GHz, 16GB RAM computer.

6.1. Data sets and configuration

In the experimental section, we use randomly generated datasets and real social network datasets, enabling us to comprehensively evaluate the performance, robustness, and adaptability of the model. The synthetic dataset is composed of randomly generated symmetric matrices of specified dimensions, and each element has an equal probability of being 0 or 1. Table 9 presents the relevant information of the three selected datasets, including dataset name, type, number of vertices, number of edges, and average degree.

- Dataset I is Jazz, which is the collaboration network between Jazz musicians. Each node is a Jazz musician and an edge denotes that two musicians have played together in a band. The data was collected in 2003.
- Dataset II is Celegans-dir, which is the metabolic network of the roundworm Caenorhabditis elegans. Nodes are metabolites (e.g., proteins), and edges are interactions between them. Since a metabolite can interact with itself, the network contains loops. The interactions are undirected. There may be multiple interactions between any two metabolites.
- Dataset III is CSphd, which contains the collaboration relationships of 1.9k computer science PhDs.

All datasets are obtained from https://networkrepository.com/index.php.
Experiments in this work are carried out for randomly synthesized and real datasets respectively.

1) **Experiments with randomly synthesized datasets:**
Experiment 1: We conducted a performance comparison test for the three algorithms. To simulate the relation matrix of symmetric formal contexts in the real world, we used randomly generated symmetric matrices. The orders of these matrices range from 5 to 20 and from 100 to 1000. Our research goal is to explore the impact of the increase in matrix dimension and recovery degree on running time, respectively, and conduct corresponding comparative analysis on these two algorithms.
Experiment 2: To more deeply and comprehensively compare the advantages and disadvantages of the three algorithms, we carefully defined two key evaluation indicators: redundancy and average recovery degree. Our purpose is to deep explore the impact of the increase in recovery degree on these two evaluation indicators, and then conduct an accurate evaluation of these three algorithms based on this.
Evaluation metrics:

Definition 16. (***Redundancy***) Under the symmetric formal context K, if the recovery degree is $\mathscr{D}$, let M_{c} be the recovery matrix under the corresponding recovery degree, and $\mathrm{SC}^{(K)}$ represents an equiconcept reduct under $\mathscr{D}$. Define the comprehensive relation number $\mathrm{U}(\mathrm{SC}^{(K)})$ corresponding to $\mathrm{SC}^{(K)}$ as:

$$U(\mathrm{SC}^{(K)}) = \sum_{(x,x)\in U(\mathrm{SC}^{(K)})} \binom{|x|}{2}, \tag{25}$$

define the redundancy $\mathrm{P}(\mathrm{SC}^{(K)})$ as:

$$P(\mathrm{SC}^{(K)}) = 1 - \frac{N_{\mathrm{c}}}{U(\mathrm{SC}^{(K)})}. \tag{26}$$

Definition 17. (***Average Recovery Degree***) Under the symmetric formal context K, if the recovery degree is $\mathscr{D}$, let M_{f} be the reference matrix, M_{c} be the recovery matrix under the corresponding recovery degree, $\mathrm{SC}^{(K)}$ represents an equiconcept reduct under $\mathscr{D}$. Define the average recovery degree $\Omega(\mathrm{SC}^{(K)})$ as:

$$\Omega(\mathrm{SC}^{(K)}) = \frac{N_{\mathrm{c}}}{N_{\mathrm{f}} \times |\mathrm{SC}^{(K)}|}. \tag{27}$$

Intuitively, we hope that the redundancy is as small as possible. The smaller the redundancy, the fewer duplicate relations are recovered, and the result is more concise and effective. For the average recovery degree, we hope it is as large as possible. The higher the average recovery degree, the more binary relations that can be recovered by an equiconcept, showing the superiority of the result.

2) **Experiments with actual datasets:**
Experiment 3: Due to the high dimension of real datasets, the time cost of RdER is too large to obtain the running result within a certain time. Therefore, only RdER+ and RdPR are tested on large-scale datasets. The impact of the increase in recovery degree on the algorithm performance in large-scale datasets is deeply studied. The evaluation indicators are still running time, redundancy, and average recovery degree.

6.2. Experimental results and analysis

To ensure the fairness of algorithm comparison, we stipulate that the measurement of running time starts from the equiconcept reduction corresponding to the given symmetric formal context and ends when obtaining the equiconcept reduction or pseudo-equiconcept reduction that satisfies the given recovery degree. Given the randomness of algorithm steps, we conduct 10 simulations for each experiment and take the average value of the results as the result value of the experiment, thereby reducing experimental errors and ensuring the fairness of the results. Before delving into a detailed discussion of the experimental results, a brief introduction to the algorithms used for comparison is provided.

- **RdER** is an algorithm that selects appropriate equiconcepts under the conditions of a given set of equiconcepts and recovery degree, so as to make the average recovery degree as large as possible.
- **RdER+** by obtaining a certain number of equiconcepts as a base to avoid repeated merging, which make the selection process faster, thereby reducing the running time.
- **RdPR** is a pseudo-equiconcept reduction algorithm based on a symmetric formal context, with the extents of equiconcept input and the extents of pseudo-equiconcept as output. Among them, all the extents of pseudo-equiconcepts are output in the form of binary relations.

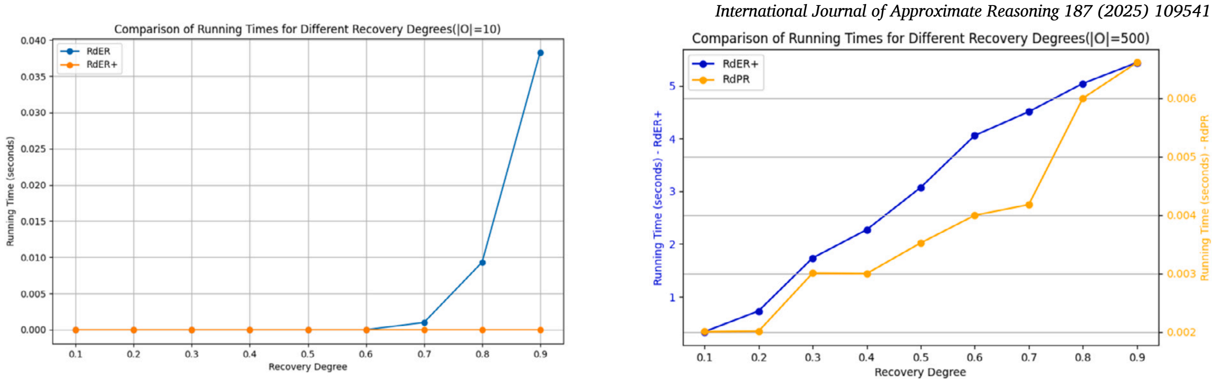
Experiment 1 (Correlation analysis of running time, degree of recovery and dimension): This experiment has two purposes. One is to explore the relationship between running time and recovery degree under the condition of specified dimensions. The second is to explore the relationship between running time and dimension construction under the condition of specified recovery degree, and compare these three algorithms. The experimental results are shown as Fig. 10 and Fig. 11.

The results illustrated in Fig. 10 evidently indicate that when the dimension remains constant, a positive correlation exists between the running time and the degree of recovery. In other words, the greater the degree of recovery, the more extended the running time will be.

Upon conducting a more in-depth comparison of the performance of different algorithms under identical circumstances, it becomes apparent that when the degree of recovery is held equal, the running time of the RdER algorithm is considerably longer than those of the other algorithms, while the running time of the RdPR algorithm is the shortest.

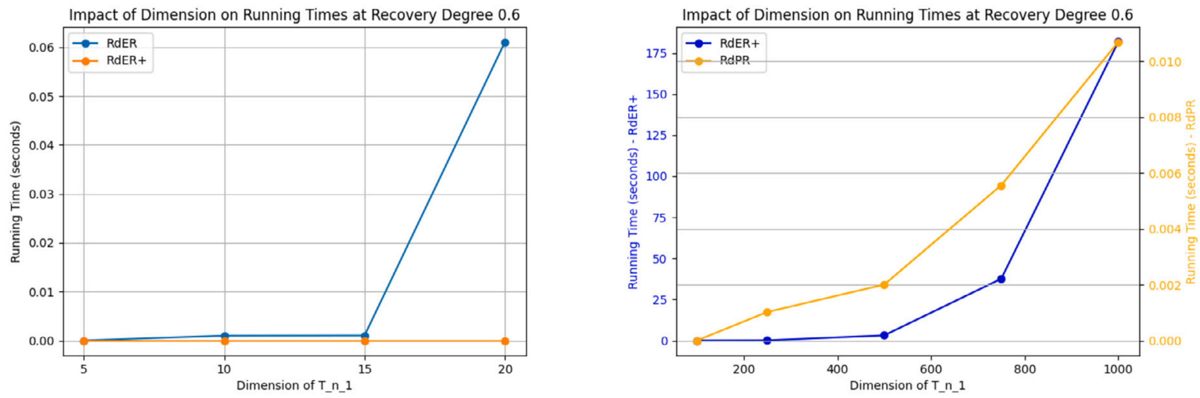
More specifically, in the case where the dimension is 10×10, once the degree of recovery surpasses the threshold of 0.6, the running time of the RdER algorithm will experience a sharp increase. In contrast, the growth trends of the running times of the other two algorithms are relatively moderate and exhibit more uniform changes.

The experimental result in Fig. 11 shows that the running time increases as the dimension increases when recovery degree is 0.6, and the growth rate of the running time of RdER is much greater than that of RdER+ and RdPR. Since in the experiment we tried to use a dimension of 25 for testing, but at this time the running time of RdER has exceeded half an hour, with poor performance.



(a) Comparison between RdER and RdER+ (b) Comparison between RdER+ and RdPR

Fig. 10. Comparison of running time in relation to the variation of recovery degree.



(a) Comparison between RdER and RdER+ (b) Comparison between RdER+ and RdPR

Fig. 11. Comparison of running time in relation to the variation of dimension.

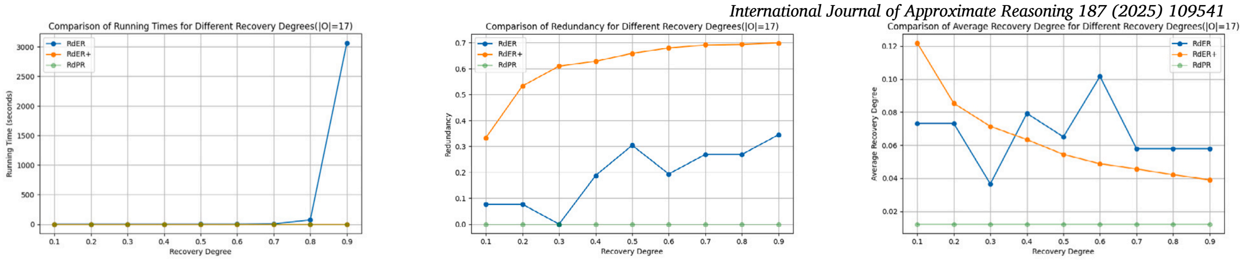
Therefore, it can be known that RdER is only applicable to small-scale matrices. The running times of RdER+ and RdPR are not sensitive to the dimension, so they are more suitable for applications of large-scale matrices.

In general, RdER is only suitable for cases with low recovery degree and small scale, while RdER+ and RdPR are more suitable for cases with high recovery degree and large scale.

Experiment 2 (Comparison of RdER, RdER+, and RdPR): This experiment aims to conduct a comprehensive performance comparison and in-depth analysis of the three algorithms RdER, RdER+, and RdPR on multiple key indicators. Specifically, starting from the three important indicators of running time, redundancy, and average recovery degree, the performance differences of different algorithms in these aspects are examined in detail. At the same time, this experiment also focuses on studying the specific impacts of the increase in recovery degree on these three algorithms. Based on the results obtained from the experiment, an objective and accurate evaluation of these three algorithms is conducted from multiple angles, and the advantages and disadvantages of each algorithm are summarized. The experimental results are shown as Fig. 12.

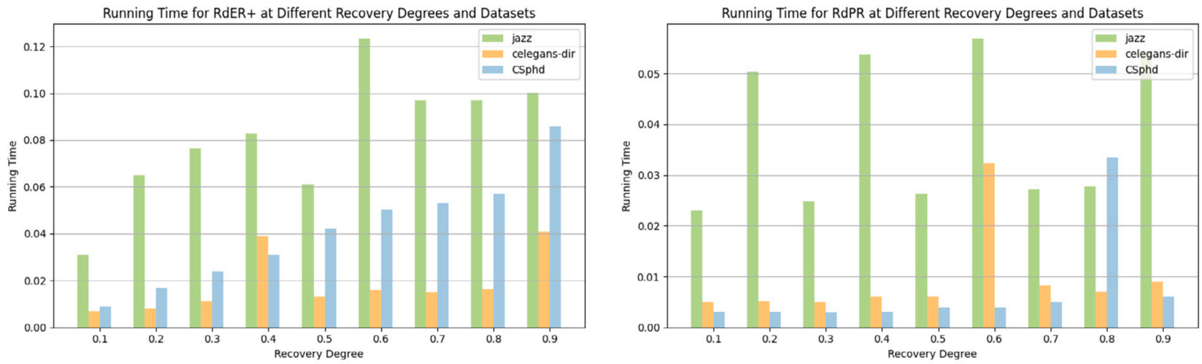
The experimental results of this time will be discussed and analyzed below:

(1) Running Time Analysis: When the dimension is 17×17, RdER+ and RdPR show excellent performance. As the recovery degree increases, the running time of these two algorithms does not increase significantly and performs well. However, the running time of RdER shows a jump growth when the recovery degree is 0.9 and exceeds half an hour, with relatively poor performance.
(2) Redundancy Analysis: When the dimension is 17×17, as the recovery degree increases, the redundancies of RdER and RdER+ both show an upward trend on the whole. And the redundancy of RdER+ is always greater than that of RdER under the same conditions, which indicates that RdER is better than RdER+ in terms of reducing redundant information. The redundancy of RdPR is always 0, which indicates that the pseudo-equiconcept reduction outperforms the equiconcept reduction in terms of redundancy and can obtain higher quality data.
(3) Average Recovery Degree Analysis: When the dimension is 17×17, as the recovery degree increases, the average recovery degree of RdER has fluctuating changes. The average recovery degree of RdER+ shows a steady downward trend, while the average recovery degree of RdPR remains unchanged at the same low value. This indicates that in terms of average recovery degree,



(a) Running Time (b) Redundancy (c) Average Recovery Degree

Fig. 12. Performance Comparisons.



(a) RdER+ (b) RdPR

Fig. 13. Running Time Comparisons.

the performance of RdER is not stable and is related to the selection of results. And the higher the recovery degree, the worse the performance of RdER+. At the same time, the performance of pseudo-equiconcept reduction is always worse than that of equiconcept reduction, which is related to the form of results.

The experimental results show that these three algorithms each have their own advantages. RdER performs well in terms of redundancy and average recovery degree, but its running time is too long and it is only applicable to small-scale and low-recovery-degree situations. RdER+ performs excellently in terms of running time, but its performance in the other two aspects is average and it is more suitable for low-recovery-degree situations. RdPR performs outstandingly in terms of running time and redundancy, but its performance in average recovery degree is poor and it is generally applicable to recovery degree.

Experiment 3 (Comparison of RdER+ and RdPR): According to the results of Experiment 2, Experiment 3 compares RdER+ and RdPR. Applying them to three real datasets respectively, it explores the relationship between three indicators and the degree of recovery. The experimental results are illustrated in Fig. 13 to Fig. 15.

The comparison results of running time are shown in Fig. 13.

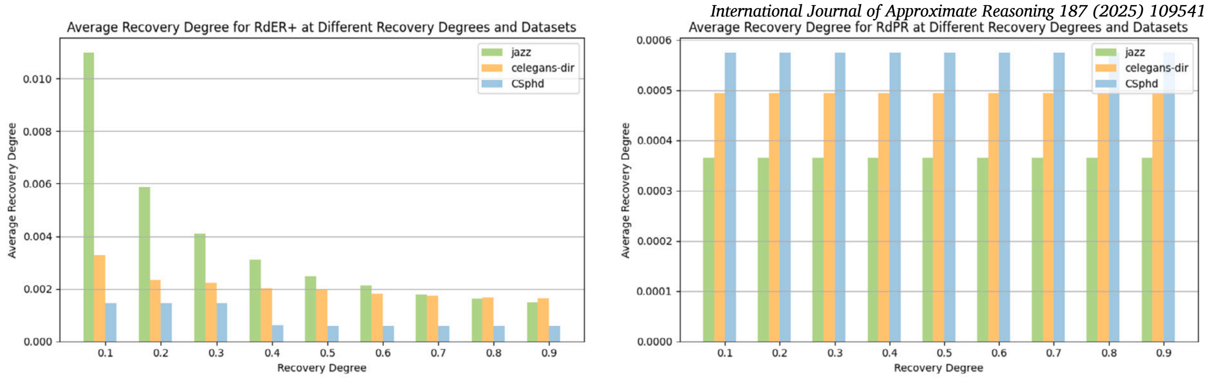
In general, the running time of RdPR is less than that of RdER+, and it has better performance. Moreover, the running time of RdPR is inversely proportional to the average degree of the data set, which indicates that the sparser the data set, the less time RdPR takes to run. However, the running time of RdER+ does not have this characteristic. As the degree of recovery increases, the running times of the two show irregular changes.

The comparison results for the average recovery degree are shown in Fig. 14.

As can be seen in the figure, the average recovery degree of RdER+ is higher than that of RdPR under the same conditions and is proportional to the average degree of the data set. The average recovery degree of RdPR is proportional to the dimension of the data set. At the same time, as the recovery degree increases, the average recovery degree of RdER+ shows a downward trend, whereas the average recovery degree of RdPR is not affected by the recovery degree.

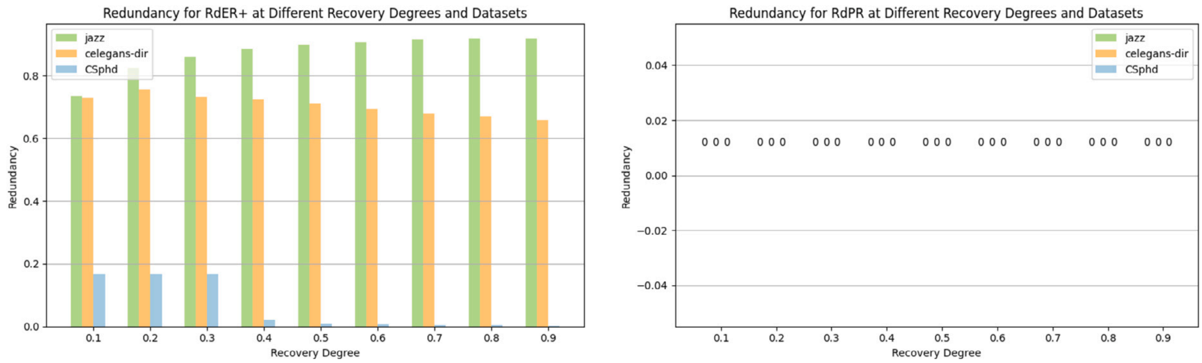
The results of the redundancy comparison are shown in Fig. 15.

As can be seen from the above figures, the redundancy of RdER+ is proportional to the average degree of the data set, while the redundancy of RdPR is always 0. This is because the pseudo-equiconcept removes duplicate binary relations based on the equiconcept, only retaining the core binary relations, which greatly reduces the redundancy of the pseudo-equiconcept. It indicates that the pseudo-equiconcept reduction shows a superior advantage in terms of redundancy.



(a) RdER+ (b) RdPR

Fig. 14. Average Recovery Degree Comparisons.



(a) RdER+ (b) RdPR

Fig. 15. Redundancy Comparisons.

7. Conclusion

Considering that the existing concept reduction algorithms can fully restore all binary relations, this paper proposes RdER, RdER+ and RdPR for partially restoring the binary relations in symmetric formal contexts. Meanwhile, we compare the performance of these three algorithms on ten artificial datasets and three real-world datasets. The experimental results demonstrate their performances in terms of running time, redundancy, and average recovery degree. Based on the experimental results, we analyze the applicability of these algorithms.

In the experiment, we found that the problem addressed in this paper has similar properties to the NP problem. In RdER, although we can ensure the optimality of each step of the calculation result, due to the high time complexity of the overall algorithm, we cannot guarantee the global optimality of the final solution. In contrast, although the improved RdER+ significantly reduces the time complexity, the optimality of its result cannot be fully guaranteed. Therefore, how to reduce the time complexity while ensuring the optimality of the result has become the core issue that needs to be further discussed in the future.

CRediT authorship contribution statement

Junyu Bu: Writing – original draft, Methodology, Investigation. **Fei Hao:** Writing – original draft, Supervision, Methodology, Funding acquisition, Conceptualization. **Huilin Fan:** Investigation, Formal analysis. **Ling Wei:** Writing – review & editing, Validation. **Shuanggen Liu:** Writing – review & editing, Software, Resources. **Sergei O. Kuznetsov:** Writing – review & editing.

Declaration of competing interest

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

Acknowledgements

This work was funded in part by the National Natural Science Foundation of China (Grant Nos. 62477029, 61702317, 12171392), the Ministry of Education Humanities and Social Sciences Research Youth Fund Project (Grant No. 22YJCZH046), the Fundamental Research Funds for the Central University (Grant No. GK202505028), the Shaanxi Fundamental Science Research Project for Mathematics and Physics (Grant No. 23JSZ008).

Data availability

All datasets can be obtained from https://networkrepository.com/index.php.

References

[1] R. Wille, Restructuring lattice theory: an approach based on hierarchies of concepts, in: Formal Concept Analysis: 7th International Conference, ICFCA 2009 Darmstadt, Germany, May 21-24, 2009 Proceedings 7, Springer, 2009, pp. 314–339.
[2] Y. Eslami, M. Lezoche, P. Kalitine, S. Ashouri, How the cooperative cyber physical enterprise information systems (ccpeis) improve the semantic interoperability in the domain of industry 4.0 through the knowledge formalization, IFAC-PapersOnLine 54 (1) (2021) 924–929.
[3] L. Zárate, M.W. Rodrigues, S.M. Dias, C. Nobre, M. Song, Scibr-m: a method to map the evolution of scientific interest-a case study in educational data mining, Library Hi Tech 41 (6) (2023) 1694–1724.
[4] A. Jaffal, B. Le Grand, M. Kirsch-Pinheiro, Refinement strategies for correlating context and user behavior in pervasive information systems, Proc. Comput. Sci. 52 (2015) 1040–1046.
[5] M.W. Rodrigues, L.E. Zárate, Orgbr-m: a method to assist in organizing bibliographic material based on formal concept analysis—a case study in educational data mining, Int. J. Digit. Libr. 21 (4) (2020) 423–448.
[6] Y. Yao, Three-way granular computing, rough sets, and formal concept analysis, Int. J. Approx. Reason. 116 (2020) 106–125.
[7] R. Ren, L. Wei, The attribute reductions of three-way concept lattices, Knowl.-Based Syst. 99 (2016) 92–102.
[8] T. Qian, L. Wei, J. Qi, Constructing three-way concept lattices based on apposition and subposition of formal contexts, Knowl.-Based Syst. 116 (2017) 39–48.
[9] M. Li, G. Wang, Approximate concept construction with three-way decisions and attribute reduction in incomplete contexts, Knowl.-Based Syst. 91 (2016) 165–178.
[10] X. Zhao, D. Miao, H. Fujita, Variable-precision three-way concepts in l-contexts, Int. J. Approx. Reason. 130 (2021) 107–125.
[11] F. Hao, G. Min, Z. Pei, D.-S. Park, L.T. Yang, k-clique community detection in social networks based on formal concept analysis, IEEE Syst. J. 11 (1) (2017) 250–259.
[12] F. Hao, D.-S. Park, G. Min, Y.-S. Jeong, J.-H. Park, k-cliques mining in dynamic social networks based on triadic formal concept analysis, Neurocomputing 209 (2016) 57–66.
[13] J.D. Kim, S.H. Hwang, D.K. Baik, Fca-based data analysis for discovering association rules in social network service, Appl. Mech. Mater. 764 (2015) 910–914.
[14] U. Priss, Modelling conceptual schemata with formal concept analysis, 2021, pp. 7–18.
[15] M. Arispe, M. Tasnim, D. Graux, F. Orlandi, D. Collarana, Verbalizing the evolution of knowledge graphs with formal concept analysis, 2020, pp. 72–79.
[16] L. Cao, L. Wei, J. Qi, Concept reduction preserving binary relations, Pattern Recognit. Artif. Intell. 31 (06) (2018) 516–524, https://doi.org/10.16451/j.cnki.issn1003-6059.201806004.
[17] L. Wei, L. Cao, J. Qi, W. Zhang, Concept reduction and concept characteristics in formal concept analysis, Sci. Sin. Inform. 50 (12) (2020) 1817–1833.
[18] L. Wei, S. Zhao, J. Qi, Symmetric formal context and its concept reduct, J. Northwest Univ. Nat. Sci. 53 (05) (2023) 794–802, https://doi.org/10.16152/j.cnki.xdxbzr.2023-05-011.
[19] G. Stacey, T. Stickley, Recovery as a threshold concept in mental health nurse education, Nurse Educ. Today 32 (5) (2012) 534–539.
[20] A.P. Sullivan, Inference and evidence in archaeology: a discussion of the conceptual problems, in: Advances in Archaeological Method and Theory, Elsevier, 1978, pp. 183–222.
[21] S. Alexander, J. Aronson, O. Whaley, D. Lamb, The relationship between ecological restoration and the ecosystem services concept, Ecol. Soc. 21 (1) (2016).
[22] H. Fan, F. Hao, L. Zhang, J. Li, L. Guo, W. Xu, S.O. Kuznetsov, V. Loia, Binary relations-preserving incremental pseudo-equiconcept reduction for symmetric formal context, Expert Syst. Appl. (2025) 127086.
[23] J. Zhao, R. Wan, D. Miao, B. Zhang, Rule acquisition of three-way semi-concept lattices in formal decision context, CAAI Trans. Intell. Technol. 9 (2) (2024) 333–347.
[24] J. Huang, Y. Lin, J. Li, Rule reductions of decision formal context based on mixed information, Appl. Intell. 53 (12) (2023) 15459–15475.
[25] G. Liu, Y. Xie, X. Gao, Three-way reduction for formal decision contexts, Inf. Sci. 615 (2022) 39–57.
[26] S. Zhang, Decision implication-based knowledge representation and reasoning within incomplete fuzzy formal context, Int. J. Fuzzy Syst. (2024) 1–16.
[27] C. Zhang, E.C. Tsang, W. Xu, Y. Lin, L. Yang, J. Wu, Dynamic updating variable precision three-way concept method based on two-way concept-cognitive learning in fuzzy formal contexts, Inf. Sci. 655 (2024) 119818.
[28] X. Liu, J. Dai, J. Chen, C. Zhang, A fuzzy α-similarity relation-based attribute reduction approach in incomplete interval-valued information systems, Appl. Soft. Comput. 109 (107593) (2021) 1568–4946.
[29] M. Hu, E.C. Tsang, Y. Guo, Q. Zhang, D. Chen, W. Xu, A novel approach to concept-cognitive learning in interval-valued formal contexts: a granular computing viewpoint, Int. J. Mach. Learn. Cybern. (2022) 1–16.
[30] Y. Wang, Z. Wang, L. Wei, R. Ren, Compound conflict degree analysis of many-valued conflict situation, J. Northwest Univ. Nat. Sci. 52 (05) (2022) 805–812, https://doi.org/10.16152/j.cnki.xdxbzr.2022-05-008.
[31] W. Zhang, L. Wei, J. Qi, Theory and method of attribute reduction for concept lattice, Sci. China, Ser. F, Inf. Sci. 06 (2005) 628–639.
[32] W.-Z. Wu, Y. Leung, J.-S. Mi, Granular computing and knowledge reduction in formal contexts, IEEE Trans. Knowl. Data Eng. 21 (10) (2008) 1461–1474.
[33] M. Liu, P. Zhu, Fuzzy object-induced network three-way concept lattice and its attribute reduction, Int. J. Approx. Reason. 173 (2024) 109251.
[34] X. Wang, W. Zhang, Knowledge Reduction in Concept Lattices Based on Irreducible Elements, Springer, 2009, pp. 128–142.
[35] X.-x. Xie, J.-j. Li, D.-x. Chen, R.-d. Lin, Concept reduction of preserving binary relations based on Boolean matrix, J. Shandong Univ. Nat. Sci. 55 (05) (2020) 32–45.
[36] Q. Wang, J. Zheng, S. Hu, Concept characteristic and concept reduction of attribute-oriented concept lattice, J. Sci. Teach. Coll. Univ. 44 (05) (2024) 35–39.
[37] R. Ren, L. Wei, J. Li, The three-way approximate concept reduction of incomplete formal contexts, Pure Appl. Math. 40 (01) (2024) 77–89.
[38] J.L. Li, X. Li, X. Wang, W. Wu, Reduction of triadic concepts based on triadic factor analysis, J. Nanjing Univ. Nat. Sci. 56 (04) (2020) 480–493, https://doi.org/10.13232/j.cnki.jnju.2020.04.006.
[39] T. Zhang, S. Li, Z. Xue, X. Lu, Fast equiconcept search on graph formal context by attribute topology, IEEE Trans. Netw. Sci. Eng. (2023).