Information Sciences 346–347 (2016) 236–260

Contents lists available at ScienceDirect

Information Sciences

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

Knowledge reduction of dynamic covering decision information systems when varying covering cardinalities

Guangming Lang[a,b,c], Duoqian Miao[a,c,*], Tian Yang[d], Mingjie Cai[e]

[a] Department of Computer Science and Technology, Tongji University, Shanghai 201804, PR China
[b] School of Mathematics and Statistics, Changsha University of Science and Technology, Changsha, Hunan 410114, PR China
[c] The Key Laboratory of Embedded System and Service Computing, Ministry of Education, Tongji University, Shanghai 201804, PR China
[d] College of Science, Central South University of Forestry and Technology, Changsha, Hunan 410004, PR China
[e] College of Mathematics and Econometrics, Hunan University, Changsha, Hunan 410082, PR China

A R T I C L E I N F O

Article history:
Received 1 June 2015
Revised 1 December 2015
Accepted 27 January 2016
Available online 10 February 2016

Keywords:
Characteristic matrix
Dynamic covering approximation space
Dynamic covering decision information system
Rough set

A B S T R A C T

In covering-based rough set theory, non-incremental approaches are time-consuming for performing knowledge reduction of dynamic covering decision information systems when the cardinalities of coverings change as a result of object immigration and emigration. Because computing approximations of sets is an important step for knowledge reduction of dynamic covering decision information systems, efficient approaches to calculating the second and sixth lower and upper approximations of sets using the type-1 and type-2 characteristic matrices, respectively, are essential. In this paper, we provide incremental approaches to computing the type-1 and type-2 characteristic matrices of dynamic coverings whose cardinalities vary with the immigration and emigration of objects. We also design incremental algorithms to compute the second and sixth lower and upper set approximations. Experimental results demonstrate that the incremental approaches effectively improve the efficiency of set approximation computation. Finally, we employ several examples to illustrate the feasibility of the incremental approaches for knowledge reduction of dynamic covering decision information systems when increasing the cardinalities of coverings.

1. Introduction

Since covering rough set theory was introduced by Zakowski in 1983, it has become a powerful mathematical tool for studying knowledge reduction of covering information systems [2,6–9,12,13,25,28,31–33,38,39,43,45,46,49,54,58,62–65,68,75–77]. In terms of theory, covering-based rough set theory has been combined with fuzzy sets, lattice theory, and other theories. In particular, different types of approximation operators have been proposed; their basic properties, including the relationships among them, have been investigated for knowledge reduction of covering information systems. In terms of applications, covering-based rough set theory has been employed to discover association rules from covering information systems for making decisions; the number of application domains is increasing with the development of covering-based rough set theory.

* Corresponding author at: Department of Computer Science and Technology, Tongji University, Shanghai 201804, PR China. Tel./fax: +86 21 69585800.
E-mail addresses: langguangming1984@tongji.edu.cn (G. Lang), dqmiao@tongji.edu.cn (D. Miao), math-yangtian@126.com (T. Yang), cmjlong@163.com (M. Cai).

http://dx.doi.org/10.1016/j.ins.2016.01.099
0020-0255/

Knowledge reduction of covering information systems is commonly transformed into computing approximations of sets. There are over 20 pairs of lower and upper approximation operators that are classified into different categories according to the criteria for covering approximation spaces. For example, Yao and Yao [69] classified all approximation operators into three types with respect to the interpretations of rough set approximations as follows: element-based approximation operators, granule-based approximation operators, and subsystem-based approximation operators; they are also classified into dual and non-dual operators with respect to the duality. To the best of our knowledge, the second and sixth lower and upper approximation operators in covering approximation spaces are typical examples of granule-based and element-based approximation operators, respectively, as well as dual operators, and they [17,18,30,47,61,73] have recently been attracting more attention with respect to the matrix view. For instance, Liu [30] proposed a new matrix view of rough set theory. Concretely, he presented a matrix representing an equivalence relation and redefined a pair of lower and upper approximation operators using the matrix representation in an approximation space. He also provided a fuzzy matrix representing a fuzzy equivalence relation and redefined the pair of lower and upper approximation operators for fuzzy sets using the matrix representation in a fuzzy approximation space. Wang et al. [61] presented the concepts of type-1 and type-2 characteristic matrices of coverings. They transformed the set approximation computation into products of the type-1 and type-2 characteristic matrices and the characteristic function of the set in covering approximation spaces. Zhang et al. [73] proposed the matrix characterizations of the lower and upper approximations for set-valued information systems and presented incremental approaches to updating the relation matrix to compute lower and upper approximations with dynamic attribute variation in set-valued information systems.

Incremental approaches [1,11,14–16,44,50] have emerged as effective methods for processing dynamic information, and knowledge reduction of dynamic information systems [3–5,17–24,26,27,29,34–37,40–42,48,52,53,59,60,66,67,72–74] has attracted many researchers. In practical situations, there exist a great deal of covering information systems such as incomplete information systems, and knowledge reduction of covering information systems is an important application of covering-based rough sets. Nowadays, matrix-based knowledge reduction of information systems [51,55–57,61,70] is increasingly adopted with the development of computer technology. For example, Skowron and Rauszer [51] provided a logical and theoretical foundation for knowledge reduction of information systems based on discernibility matrices. Tan et al. [55] proposed matrix-based methods for computing set approximations and reducts of covering decision information systems. Concretely, they introduced some matrices and matrix operations for computing the positive regions of covering decision information systems and employed the minimal and maximal descriptions to construct a new discernibility matrix for attribute reduction of covering information systems. Tan et al. [56] also provided fast approaches to knowledge reduction of covering information systems by employing novel matrix operations. Wang et al. [57] provided a novel method for attribute reduction of covering information systems. In particular, they developed a judgment theorem and a discernibility matrix for consistent and inconsistent covering decision systems, and presented a heuristic algorithm to find a subset of attributes to approximate a minimal reduct based on discernibility matrices. Yao and Zhao [70] proposed elementary matrix simplification operations to transform a matrix into a simpler form using heuristic reduct construction algorithms with respect to the ordering of attributes.

In practice, dynamic covering approximation spaces are time-varying. Correspondingly, dynamic covering information systems are time-varying, and the effective knowledge reduction of dynamic covering approximation spaces and covering information systems has become a significant challenge. Knowledge reduction of dynamic covering information systems is transformed into the construction of the set approximations. It is therefore necessary to investigate the computation of set approximations in dynamic covering approximation spaces. To address this, Lang et al. [17,18] presented incremental approaches to computing the second and sixth lower and upper approximations of sets in dynamic covering approximation spaces and performed knowledge reduction of dynamic covering information systems with variations of object sets. However, they focused on dynamic covering approximation spaces whose dynamic covering cardinalities remain unchanged with varying object sets. In reality, there are many dynamic covering approximation spaces whose dynamic covering cardinalities increase or decrease with the variation of object sets. Thus, computing the second and sixth lower and upper approximations of sets for knowledge reduction of dynamic covering information systems using the type-1 and type-2 characteristic matrices in the dynamic covering approximation spaces will enrich covering-based rough set theory from the matrix view.

The purpose of this paper is to investigate knowledge reduction of dynamic covering decision information systems while changing the cardinalities of coverings. First, we study the properties of dynamic coverings whose cardinalities increase and decrease because of object immigration and emigration, respectively, and present incremental approaches to computing the type-1 and type-2 characteristic matrices of dynamic coverings for the construction of set approximations. We also show four examples to demonstrate the construction of the second and sixth lower and upper approximations of sets in dynamic covering approximation spaces. Second, we provide incremental algorithms for constructing the second and sixth lower and upper approximations of sets using the type-1 and type-2 characteristic matrices, respectively. We also compare the computational complexities of the incremental algorithms with those of non-incremental algorithms. Third, we apply the incremental algorithms to large-scale dynamic covering approximation spaces and employ the experimental results to demonstrate that the incremental algorithms effectively compute the second and sixth lower and upper approximations of sets in dynamic covering approximation spaces. We show several examples to illustrate the knowledge reduction of dynamic covering decision information systems while increasing the cardinalities of coverings.

The remaining of this paper is organized as follows. Section 2 briefly reviews the basic concepts of covering-based rough set theory. In Section 3, we introduce incremental approaches to computing the type-1 and type-2 characteristic matrices of dynamic coverings. In Section 4, we design incremental algorithms to calculate the second and sixth lower and upper approximations of sets. In Section 5, the experimental results demonstrate that incremental approaches improve the effectiveness of computing approximations of sets. In Section 6, we provide examples showing the process of performing knowledge reduction of dynamic covering decision information systems when increasing the cardinalities of coverings. Concluding remarks are presented in Section 7.

2. Preliminaries

In this section, we briefly review some concepts of covering-based rough sets.

Definition 2.1 [71]**.** Let U be a finite universe of discourse, and $\mathscr{C}$ a family of subsets of U. Then $\mathscr{C}$ is called a covering of U if none of elements of $\mathscr{C}$ is empty and $\bigcup\{C|C\in\mathscr{C}\}=U$. Furthermore, $(U,\mathscr{C})$ is referred to as a covering approximation space.

According to Definition 2.1, a covering is an extension of a partition based on an equivalence relation, and a covering approximation space is a generalization of Pawlak's approximation space.

Definition 2.2 [61]**.** Let $U=\{x_1,x_2,\ldots,x_n\}$ be a finite universe, and $\mathscr{C}=\{C_1,C_2,\ldots,C_m\}$ a covering of U, and $N(x)=\bigcap\{C_i|x\in C_i\in\mathscr{C}\}$ for $x\in U$. For any $X\subseteq U$, the second and sixth upper and lower approximations of X with respect to $\mathscr{C}$ are defined as follows:

(1) $SH_{\mathscr{C}}(X)=\bigcup\{C\in\mathscr{C}|C\cap X\neq\emptyset\},\ SL_{\mathscr{C}}(X)=[SH_{\mathscr{C}}(X^c)]^c$;
(2) $XH_{\mathscr{C}}(X)=\{x\in U|N(x)\cap X\neq\emptyset\},\ XL_{\mathscr{C}}(X)=\{x\in U|N(x)\subseteq X\}$.

According to Definition 2.2, the second and sixth lower and upper approximation operators are typical examples of granule-based and element-based approximation operators, respectively, and are also dual operators. Furthermore, the second and sixth lower and upper approximation operators are typical examples of approximation operators constructed on the original coverings and induced coverings, respectively.

Definition 2.3 [61]**.** Let $U=\{x_1,x_2,\ldots,x_n\}$ be a finite universe, $\mathscr{C}=\{C_1,C_2,\ldots,C_m\}$ a family of subsets of U, and $M_{\mathscr{C}}=(a_{ij})_{n\times m}$, where $a_{ij}=\begin{cases}1, & x_i\in C_j;\\ 0, & x_i\notin C_j.\end{cases}$ Then $M_{\mathscr{C}}$ is called a matrix representation of $\mathscr{C}$.

We also have the characteristic function $\mathcal{X}_X=[a_1\ \ a_2\ \ .\ \ .\ \ .\ \ a_n]^T$ for $X\subseteq U$, where $a_i=\begin{cases}1, & x_i\in X;\\ 0, & x_i\notin X.\end{cases}$.

Definition 2.4 [61]**.** Let $\mathscr{C}$ be a covering of the universe U, $A=(a_{ij})_{n\times m}$ and $B=(b_{ij})_{m\times p}$ Boolean matrices, $A\odot B=(c_{ij})_{n\times p}$, where $c_{ij}=\bigwedge_{k=1}^{m}(b_{kj}-a_{ik}+1)$. Then

(1) $\Gamma(\mathscr{C})=M_{\mathscr{C}}\bullet M_{\mathscr{C}}^T=(d_{ij})_{n\times n}$ is called the type-1 characteristic matrix of $\mathscr{C}$, where $d_{ij}=\bigvee_{k=1}^{m}(a_{ik}\cdot a_{jk})$, and $M_{\mathscr{C}}\bullet M_{\mathscr{C}}^T$ is the boolean product of $M_{\mathscr{C}}$ and its transpose $M_{\mathscr{C}}^T$;
(2) $\prod(\mathscr{C})=M_{\mathscr{C}}\odot M_{\mathscr{C}}^T=(e_{ij})_{n\times n}$ is called the type-2 characteristic matrix of $\mathscr{C}$.

Wang et al. axiomatized the second and sixth lower and upper approximation operators equivalently using the type-1 and type-2 characteristic matrices of $\mathscr{C}$.

Definition 2.5 [61]**.** Let $U=\{x_1,x_2,\ldots,x_n\}$ be a finite universe, $\mathscr{C}=\{C_1,C_2,\ldots,C_m\}$ a covering of U, and $\mathcal{X}_X$ the characteristic function of X in U. Then

(1) $\mathcal{X}_{SH(X)}=\Gamma(\mathscr{C})\bullet\mathcal{X}_X,\ \mathcal{X}_{SL(X)}=\Gamma(\mathscr{C})\odot\mathcal{X}_X$; (2) $\mathcal{X}_{XH(X)}=\prod(\mathscr{C})\bullet\mathcal{X}_X,\ \mathcal{X}_{XL(X)}=\prod(\mathscr{C})\odot\mathcal{X}_X$.

In [30], Liu also presented a pair of lower and upper approximation operators using the matrix representation of an equivalence relation in approximation spaces, which is similar to the second lower and upper approximation operators using the type-1 characteristic matrix in covering approximation spaces.

Definition 2.6 [17]**.** Let $(U,\mathscr{D}\cup U/d)$ be a covering decision information system, where $\mathscr{D}=\{\mathscr{C}_i|i\in I\}$, $U/d=\{D_i|i\in J\}$, I and J are indexed sets. Then $\mathscr{P}\subseteq\mathscr{D}$ is called a type-1 reduct of $(U,\mathscr{D}\cup U/d)$ if it satisfies

(1) $\Gamma(\mathscr{D})\bullet\mathcal{X}_{D_i}=\Gamma(\mathscr{P})\bullet\mathcal{X}_{D_i},\Gamma(\mathscr{D})\odot\mathcal{X}_{D_i}=\Gamma(\mathscr{P})\odot\mathcal{X}_{D_i},\forall i\in J$;
(2) $\Gamma(\mathscr{D})\bullet\mathcal{X}_{D_i}\neq\Gamma(\mathscr{P}')\bullet\mathcal{X}_{D_i},\Gamma(\mathscr{D})\odot\mathcal{X}_{D_i}\neq\Gamma(\mathscr{P}')\odot\mathcal{X}_{D_i},\forall\mathscr{P}'\subset\mathscr{P}$.

Definition 2.7 [17]**.** Let $(U,\mathscr{D}\cup U/d)$ be a covering decision information system, where $\mathscr{D}=\{\mathscr{C}_i|i\in I\}$, $U/d=\{D_i|i\in J\}$, I and J are indexed sets. Then $\mathscr{P}\subseteq\mathscr{D}$ is called a type-2 reduct of $(U,\mathscr{D}\cup U/d)$ if it satisfies

(1) $\prod(\mathscr{D})\bullet\mathcal{X}_{D_i}=\prod(\mathscr{P})\bullet\mathcal{X}_{D_i},\prod(\mathscr{D})\odot\mathcal{X}_{D_i}=\prod(\mathscr{P})\odot\mathcal{X}_{D_i},\forall i\in J$;
(2) $\prod(\mathscr{D})\bullet\mathcal{X}_{D_i}\neq\prod(\mathscr{P}')\bullet\mathcal{X}_{D_i},\prod(\mathscr{D})\odot\mathcal{X}_{D_i}\neq\prod(\mathscr{P}')\odot\mathcal{X}_{D_i},\forall\mathscr{P}'\subset\mathscr{P}$.

3. Update approximations of sets with object immigration and emigration

In this section, we introduce incremental approaches to computing the second and sixth lower and upper approximation of sets with object immigration and emigration.

Definition 3.1. Let $(U,\mathscr{C})$ and $(U^+,\mathscr{C}^+)$ be covering approximation spaces, where $U=\{x_1,x_2,\ldots,x_n\}$, $U^+=\{x_1,x_2,\ldots,x_n,x_{n+1},x_{n+2},\ldots,x_{n+t}\}(t\geq 2)$, $\mathscr{C}=\{C_1,C_2,\ldots,C_m\}$, $\mathscr{C}^+=\{C_1^+,C_2^+,\ldots,C_m^+,C_{m+1}^+,C_{m+2}^+,\ldots,C_{m+l}^+\}(l\geq 2)$, where $C_i^+=C_i\cup\Delta C_i$ $(1\leq i\leq m)$, and $\Delta C_i\subseteq\{x_{n+1},x_{n+2},\ldots,x_{n+t}\}$. Then $(U^+,\mathscr{C}^+)$ is called a dynamic covering approximation space of $(U,\mathscr{C})$.

While there are several types of dynamic covering approximation spaces when adding more objects, we only discuss the type shown in Definition 3.1 for simplicity.

Example 3.2. Let $(U,\mathscr{C})$ be a covering approximation space, where $U=\{x_1,x_2,x_3,x_4\}$, $\mathscr{C}=\{C_1,C_2,C_3\}$, $C_1=\{x_1,x_4\}$, $C_2=\{x_1,x_2,x_4\}$, and $C_3=\{x_3,x_4\}$. By adding $\{x_5,\ x_6\}$ into U, we get a dynamic covering approximation space $(U^+,\mathscr{C}^+)$ of $(U,\mathscr{C})$, where $U^+=\{x_1,x_2,x_3,x_4,x_5,x_6\}$, $\mathscr{C}^+=\{C_1^+,C_2^+,C_3^+,C_4^+,C_5^+\}$, $C_1^+=\{x_1,x_4,x_5\}$, $C_2^+=\{x_1,x_2,x_4,x_5\}$, $C_3^+=\{x_3,x_4\}$, $C_4^+=\{x_3,x_5,x_6\}$, and $C_5^+=\{x_1,x_6\}$.

In what follows, we show how to construct $\Gamma(\mathscr{C}^+)$ based on $\Gamma(\mathscr{C})$. For convenience, we denote $M_{\mathscr{C}}=(a_{ij})_{n\times m}$, $M_{\mathscr{C}^+}=(a_{ij})_{(n+t)\times(m+l)}$, $\Gamma(\mathscr{C})=(b_{ij})_{n\times n}$, and $\Gamma(\mathscr{C}^+)=(c_{ij})_{(n+t)\times(n+t)}$.

Theorem 3.3. *Let $(U^+,\mathscr{C}^+)$ be a dynamic covering approximation space of $(U,\mathscr{C})$, $\Gamma(\mathscr{C})$ and $\Gamma(\mathscr{C}^+)$ the type-1 characteristic matrices of $\mathscr{C}$ and $\mathscr{C}^+$, respectively. Then*

$$\Gamma(\mathscr{C}^+)=\begin{bmatrix}\Gamma(\mathscr{C}) & 0\\ 0 & 0\end{bmatrix}\bigvee\begin{bmatrix}\triangle_1(\Gamma(\mathscr{C})) & (\triangle_2(\Gamma(\mathscr{C})))^T\\ \triangle_2(\Gamma(\mathscr{C})) & \triangle_3(\Gamma(\mathscr{C}))\end{bmatrix},$$

where

$$\triangle_1(\Gamma(\mathscr{C}))=\begin{bmatrix}a_{1(m+1)} & a_{2(m+1)} & . & . & . & a_{n(m+1)}\\ a_{1(m+2)} & a_{2(m+2)} & . & . & . & a_{n(m+2)}\\ . & . & . & . & . & .\\ . & . & . & . & . & .\\ . & . & . & . & . & .\\ a_{1(m+l)} & a_{2(m+l)} & . & . & . & a_{n(m+l)}\end{bmatrix}^T\bullet\begin{bmatrix}a_{1(m+1)} & a_{2(m+1)} & . & . & . & a_{n(m+1)}\\ a_{1(m+2)} & a_{2(m+2)} & . & . & . & a_{n(m+2)}\\ . & . & . & . & . & .\\ . & . & . & . & . & .\\ . & . & . & . & . & .\\ a_{1(m+l)} & a_{2(m+l)} & . & . & . & a_{n(m+l)}\end{bmatrix},$$

$$\triangle_2(\Gamma(\mathscr{C}))=\begin{bmatrix}a_{(n+1)1} & a_{(n+1)2} & . & . & . & a_{(n+1)(m+l)}\\ a_{(n+2)1} & a_{(n+2)2} & . & . & . & a_{(n+2)(m+l)}\\ . & . & . & . & . & .\\ . & . & . & . & . & .\\ . & . & . & . & . & .\\ a_{(n+t)1} & a_{(n+t)2} & . & . & . & a_{(n+t)(m+l)}\end{bmatrix}\bullet\begin{bmatrix}a_{11} & a_{12} & . & . & . & a_{1(m+l)}\\ a_{21} & a_{22} & . & . & . & a_{2(m+l)}\\ . & . & . & . & . & .\\ . & . & . & . & . & .\\ . & . & . & . & . & .\\ a_{n1} & a_{n2} & . & . & . & a_{n(m+l)}\end{bmatrix}^T,$$

$$\triangle_3(\Gamma(\mathscr{C}))=\begin{bmatrix}a_{(n+1)1} & a_{(n+1)2} & . & . & . & a_{(n+1)(m+l)}\\ a_{(n+2)1} & a_{(n+2)2} & . & . & . & a_{(n+2)(m+l)}\\ . & . & . & . & . & .\\ . & . & . & . & . & .\\ . & . & . & . & . & .\\ a_{(n+t)1} & a_{(n+t)2} & . & . & . & a_{(n+t)(m+l)}\end{bmatrix}\bullet\begin{bmatrix}a_{(n+1)1} & a_{(n+1)2} & . & . & . & a_{(n+1)(m+l)}\\ a_{(n+2)1} & a_{(n+2)2} & . & . & . & a_{(n+2)(m+l)}\\ . & . & . & . & . & .\\ . & . & . & . & . & .\\ . & . & . & . & . & .\\ a_{(n+t)1} & a_{(n+t)2} & . & . & . & a_{(n+t)(m+l)}\end{bmatrix}^T.$$

Proof. By Definitions 2.4 and 3.1, we get $\Gamma(\mathscr{C})$ and $\Gamma(\mathscr{C}^+)$ as follows:

$$\begin{aligned}\Gamma(\mathscr{C}) &= M_{\mathscr{C}}\bullet M_{\mathscr{C}}^T\\ &=\begin{bmatrix}a_{11} & a_{12} & . & . & . & a_{1m}\\ a_{21} & a_{22} & . & . & . & a_{2m}\\ . & . & . & . & . & .\\ . & . & . & . & . & .\\ . & . & . & . & . & .\\ a_{n1} & a_{n2} & . & . & . & a_{nm}\end{bmatrix}\bullet\begin{bmatrix}a_{11} & a_{12} & . & . & . & a_{1m}\\ a_{21} & a_{22} & . & . & . & a_{2m}\\ . & . & . & . & . & .\\ . & . & . & . & . & .\\ . & . & . & . & . & .\\ a_{n1} & a_{n2} & . & . & . & a_{nm}\end{bmatrix}^T\\ &=\begin{bmatrix}b_{11} & b_{12} & . & . & . & b_{1n}\\ b_{21} & b_{22} & . & . & . & b_{2n}\\ . & . & . & . & . & .\\ . & . & . & . & . & .\\ . & . & . & . & . & .\\ b_{n1} & b_{n2} & . & . & . & b_{nn}\end{bmatrix},\\ \Gamma(\mathscr{C}^+) &= M_{\mathscr{C}^+}\bullet M_{\mathscr{C}^+}^T\end{aligned}$$

$$= \begin{bmatrix} a_{11} & a_{12} & . & . & . & a_{1m} & a_{1(m+1)} & . & . & . & a_{1(m+l)} \\ a_{21} & a_{22} & . & . & . & a_{2m} & a_{2(m+1)} & . & . & . & a_{2(m+l)} \\ . & . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . & . & . \\ a_{n1} & a_{n2} & . & . & . & a_{nm} & a_{n(m+1)} & . & . & . & a_{n(m+l)} \\ a_{(n+1)1} & a_{(n+1)2} & . & . & . & a_{(n+1)m} & a_{(n+1)(m+1)} & . & . & . & a_{(n+1)(m+l)} \\ . & . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . & . & . \\ a_{(n+t)1} & a_{(n+t)2} & . & . & . & a_{(n+t)m} & a_{(n+t)(m+1)} & . & . & . & a_{(n+t)(m+l)} \end{bmatrix} \bullet$$

$$\begin{bmatrix} a_{11} & a_{12} & . & . & . & a_{1m} & a_{1(m+1)} & . & . & . & a_{1(m+l)} \\ a_{21} & a_{22} & . & . & . & a_{2m} & a_{2(m+1)} & . & . & . & a_{2(m+l)} \\ . & . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . & . & . \\ a_{n1} & a_{n2} & . & . & . & a_{nm} & a_{n(m+1)} & . & . & . & a_{n(m+l)} \\ a_{(n+1)1} & a_{(n+1)2} & . & . & . & a_{(n+1)m} & a_{(n+1)(m+1)} & . & . & . & a_{(n+1)(m+l)} \\ . & . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . & . & . \\ a_{(n+t)1} & a_{(n+t)2} & . & . & . & a_{(n+t)m} & a_{(n+t)(m+1)} & . & . & . & a_{(n+t)(m+l)} \end{bmatrix}^T$$

$$= \begin{bmatrix} c_{11} & c_{12} & . & . & . & c_{1n} & c_{1(n+1)} & . & . & . & c_{1(n+t)} \\ c_{21} & c_{22} & . & . & . & c_{2n} & c_{2(n+1)} & . & . & . & c_{2(n+t)} \\ . & . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . & . & . \\ c_{n1} & c_{n2} & . & . & . & c_{nn} & c_{n(n+1)} & . & . & . & c_{n(n+t)} \\ c_{(n+1)1} & c_{(n+1)2} & . & . & . & c_{(n+1)n} & c_{(n+1)(n+1)} & . & . & . & c_{(n+1)(n+t)} \\ . & . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . & . & . \\ c_{(n+t)1} & c_{(n+t)2} & . & . & . & c_{(n+t)n} & c_{(n+t)(n+1)} & . & . & . & c_{(n+t)(n+t)} \end{bmatrix}.$$

By Definition 2.4, we have that

$$\begin{aligned} c_{11} &= \begin{bmatrix} a_{11} & a_{12} & . & . & . & a_{1m} & a_{1(m+1)} & . & . & . & a_{1(m+l)} \end{bmatrix} \bullet \\ &\quad \begin{bmatrix} a_{11} & a_{12} & . & . & . & a_{1m} & a_{1(m+1)} & . & . & . & a_{1(m+l)} \end{bmatrix}^T \\ &= \begin{bmatrix} a_{11} & a_{12} & . & . & . & a_{1m} \end{bmatrix} \bullet \begin{bmatrix} a_{11} & a_{12} & . & . & . & a_{1m} \end{bmatrix}^T \vee \\ &\quad \begin{bmatrix} a_{1(m+1)} & . & . & . & a_{1(m+l)} \end{bmatrix} \bullet \begin{bmatrix} a_{1(m+1)} & . & . & . & a_{1(m+l)} \end{bmatrix}^T \\ &= b_{11} \vee \begin{bmatrix} a_{1(m+1)} & . & . & . & a_{1(m+l)} \end{bmatrix} \bullet \begin{bmatrix} a_{1(m+1)} & . & . & . & a_{1(m+l)} \end{bmatrix}^T, \\ c_{(n+1)1} &= \begin{bmatrix} a_{(n+1)1} & a_{(n+1)2} & . & . & . & a_{(n+1)m} & a_{(n+1)(m+1)} & . & . & . & a_{(n+1)(m+l)} \end{bmatrix} \bullet \\ &\quad \begin{bmatrix} a_{11} & a_{12} & . & . & . & a_{1m} & a_{1(m+1)} & . & . & . & a_{1(m+l)} \end{bmatrix}^T \\ &= 0 \vee \begin{bmatrix} a_{(n+1)1} & a_{(n+1)2} & . & . & . & a_{(n+1)m} & a_{(n+1)(m+1)} & . & . & . & a_{(n+1)(m+l)} \end{bmatrix} \bullet \\ &\quad \begin{bmatrix} a_{11} & a_{12} & . & . & . & a_{1m} & a_{1(m+1)} & . & . & . & a_{1(m+l)} \end{bmatrix}^T, \\ c_{(n+1)(n+1)} &= \begin{bmatrix} a_{(n+1)1} & a_{(n+1)2} & . & . & . & a_{(n+1)m} & a_{(n+1)(m+1)} & . & . & . & a_{(n+1)(m+l)} \end{bmatrix} \bullet \\ &\quad \begin{bmatrix} a_{(n+1)1} & a_{(n+1)2} & . & . & . & a_{(n+1)m} & a_{(n+1)(m+1)} & . & . & . & a_{(n+1)(m+l)} \end{bmatrix}^T \\ &= 0 \vee \begin{bmatrix} a_{(n+1)1} & a_{(n+1)2} & . & . & . & a_{(n+1)m} & a_{(n+1)(m+1)} & . & . & . & a_{(n+1)(m+l)} \end{bmatrix} \bullet \\ &\quad \begin{bmatrix} a_{(n+1)1} & a_{(n+1)2} & . & . & . & a_{(n+1)m} & a_{(n+1)(m+1)} & . & . & . & a_{(n+1)(m+l)} \end{bmatrix}^T. \end{aligned}$$

Since $c_{11} \in \triangle_1(\Gamma(\mathscr{C}))$, $c_{(n+1)1} \in \triangle_2(\Gamma(\mathscr{C}))$, and $c_{(n+1)(n+1)} \in \triangle_3(\Gamma(\mathscr{C}))$, we can compute the other elements of $\triangle_1(\Gamma(\mathscr{C}))$, $\triangle_2(\Gamma(\mathscr{C}))$, and $\triangle_3(\Gamma(\mathscr{C}))$ similarly. To obtain $\Gamma(\mathscr{C}^+)$, we only need to compute $\triangle_1(\Gamma(\mathscr{C}))$, $\triangle_2(\Gamma(\mathscr{C}))$, and

$\triangle_3(\Gamma(\mathscr{C}))$ on the basis of $\Gamma(\mathscr{C})$ as follows:

$$\triangle_1(\Gamma(\mathscr{C})) = \begin{bmatrix} a_{1(m+1)} & a_{2(m+1)} & \cdot & \cdot & \cdot & a_{n(m+1)} \\ a_{1(m+2)} & a_{2(m+2)} & \cdot & \cdot & \cdot & a_{n(m+2)} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{1(m+l)} & a_{2(m+l)} & \cdot & \cdot & \cdot & a_{n(m+l)} \end{bmatrix}^T \bullet \begin{bmatrix} a_{1(m+1)} & a_{2(m+1)} & \cdot & \cdot & \cdot & a_{n(m+1)} \\ a_{1(m+2)} & a_{2(m+2)} & \cdot & \cdot & \cdot & a_{n(m+2)} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{1(m+l)} & a_{2(m+l)} & \cdot & \cdot & \cdot & a_{n(m+l)} \end{bmatrix},$$

$$\triangle_2(\Gamma(\mathscr{C})) = \begin{bmatrix} a_{(n+1)1} & a_{(n+1)2} & \cdot & \cdot & \cdot & a_{(n+1)(m+l)} \\ a_{(n+2)1} & a_{(n+2)2} & \cdot & \cdot & \cdot & a_{(n+2)(m+l)} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{(n+t)1} & a_{(n+t)2} & \cdot & \cdot & \cdot & a_{(n+t)(m+l)} \end{bmatrix} \bullet \begin{bmatrix} a_{11} & a_{12} & \cdot & \cdot & \cdot & a_{1(m+l)} \\ a_{21} & a_{22} & \cdot & \cdot & \cdot & a_{2(m+l)} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{n1} & a_{n2} & \cdot & \cdot & \cdot & a_{n(m+l)} \end{bmatrix}^T,$$

$$\triangle_3(\Gamma(\mathscr{C})) = \begin{bmatrix} a_{(n+1)1} & a_{(n+1)2} & \cdot & \cdot & \cdot & a_{(n+1)(m+l)} \\ a_{(n+2)1} & a_{(n+2)2} & \cdot & \cdot & \cdot & a_{(n+2)(m+l)} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{(n+t)1} & a_{(n+t)2} & \cdot & \cdot & \cdot & a_{(n+t)(m+l)} \end{bmatrix} \bullet \begin{bmatrix} a_{(n+1)1} & a_{(n+1)2} & \cdot & \cdot & \cdot & a_{(n+1)(m+l)} \\ a_{(n+2)1} & a_{(n+2)2} & \cdot & \cdot & \cdot & a_{(n+2)(m+l)} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{(n+t)1} & a_{(n+t)2} & \cdot & \cdot & \cdot & a_{(n+t)(m+l)} \end{bmatrix}^T.$$

Therefore, we have

$$\Gamma(\mathscr{C}^+) = \begin{bmatrix} \Gamma(\mathscr{C}) & 0 \\ 0 & 0 \end{bmatrix} \bigvee \begin{bmatrix} \triangle_1(\Gamma(\mathscr{C})) & (\triangle_2(\Gamma(\mathscr{C})))^T \\ \triangle_2(\Gamma(\mathscr{C})) & \triangle_3(\Gamma(\mathscr{C})) \end{bmatrix}.$$

□

Example 3.4 (Continued from Example 3.2). Taking $X = \{x_3, x_4, x_5\}$. By Definition 2.4, we first have that

$$\Gamma(\mathscr{C}) = M_{\mathscr{C}} \bullet M_{\mathscr{C}}^T = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix} \bullet \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}^T = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}.$$

Second, by Theorem 3.3, we get that

$$\triangle_1(\Gamma(\mathscr{C})) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}^T \bullet \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\triangle_2(\Gamma(\mathscr{C})) = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \bullet \begin{bmatrix} 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{bmatrix}^T = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix},$$

$$\triangle_3(\Gamma(\mathscr{C})) = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \bullet \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}^T = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}.$$

Third, we obtain that

$$\Gamma(\mathscr{C}^+) = (c_{ij})_{66} = \begin{bmatrix} \Gamma(\mathscr{C}) & 0 \\ 0 & 0 \end{bmatrix} \bigvee \begin{bmatrix} \triangle_1(\Gamma(\mathscr{C})) & (\triangle_2(\Gamma(\mathscr{C})))^T \\ \triangle_2(\Gamma(\mathscr{C})) & \triangle_3(\Gamma(\mathscr{C})) \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \bigvee \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}.$$

By Definition 2.5, we have that

$$\mathcal{X}_{SH(X)} = \Gamma(\mathscr{C}^+) \bullet \mathcal{X}_X = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \bullet \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}^T,$$

$$\mathcal{X}_{SL(X)} = \Gamma(\mathscr{C}^+) \odot \mathcal{X}_X = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \odot \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T.$$

Therefore, $SH(X) = \{x_1, x_2, x_3, x_4, x_5, x_6\}$ and $SL(X) = \emptyset$.

In Example 3.4, we must compute all of the elements in $\Gamma(\mathscr{C}^+)$ to construct the set approximations by Definition 2.4. In contrast, we only need to calculate elements in $\triangle_1(\Gamma(\mathscr{C}))$, $\triangle_2(\Gamma(\mathscr{C}))$, and $\triangle_3(\Gamma(\mathscr{C}))$ by Theorem 3.3. Thereby, the incremental approach is more effective to compute the second lower and upper approximations of sets.

Subsequently, we construct $\prod(\mathscr{C}^+)$ based on $\prod(\mathscr{C})$. For convenience, we denote $\prod(\mathscr{C}) = (d_{ij})_{n\times n}$ and $\prod(\mathscr{C}^+) = (e_{ij})_{(n+t)\times(n+t)}$.

Theorem 3.5. *Let $(U^+, \mathscr{C}^+)$ be a dynamic covering approximation space of $(U, \mathscr{C})$, $\prod(\mathscr{C})$ and $\prod(\mathscr{C}^+)$ the type-2 characteristic matrices of $\mathscr{C}$ and $\mathscr{C}^+$, respectively. Then*

$$\prod(\mathscr{C}^+) = \begin{bmatrix} \prod(\mathscr{C}) & 1 \\ 1 & 1 \end{bmatrix} \bigwedge \begin{bmatrix} \triangle_1(\prod(\mathscr{C})) & \triangle_3(\prod(\mathscr{C})) \\ \triangle_2(\prod(\mathscr{C})) & \triangle_4(\prod(\mathscr{C})) \end{bmatrix},$$

where

$$\triangle_1\left(\prod(\mathscr{C})\right) = \begin{bmatrix} a_{1(m+1)} & a_{2(m+1)} & \cdot & \cdot & \cdot & a_{n(m+1)} \\ a_{1(m+2)} & a_{2(m+2)} & \cdot & \cdot & \cdot & a_{n(m+2)} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{1(m+l)} & a_{2(m+l)} & \cdot & \cdot & \cdot & a_{n(m+l)} \end{bmatrix}^T \odot \begin{bmatrix} a_{1(m+1)} & a_{2(m+1)} & \cdot & \cdot & \cdot & a_{n(m+1)} \\ a_{1(m+2)} & a_{2(m+2)} & \cdot & \cdot & \cdot & a_{n(m+2)} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{1(m+l)} & a_{2(m+l)} & \cdot & \cdot & \cdot & a_{n(m+l)} \end{bmatrix},$$

$$\triangle_2\left(\prod(\mathscr{C})\right) = \begin{bmatrix} a_{(n+1)1} & a_{(n+1)2} & \cdot & \cdot & \cdot & a_{(n+1)(m+l)} \\ a_{(n+2)1} & a_{(n+2)2} & \cdot & \cdot & \cdot & a_{(n+2)(m+l)} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{(n+t)1} & a_{(n+t)2} & \cdot & \cdot & \cdot & a_{(n+t)(m+l)} \end{bmatrix} \odot \begin{bmatrix} a_{11} & a_{12} & \cdot & \cdot & \cdot & a_{1(m+l)} \\ a_{21} & a_{22} & \cdot & \cdot & \cdot & a_{2(m+l)} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{n1} & a_{n2} & \cdot & \cdot & \cdot & a_{n(m+l)} \end{bmatrix}^T,$$

$$\triangle_3\left(\prod(\mathscr{C})\right) = \begin{bmatrix} a_{11} & a_{12} & \cdot & \cdot & \cdot & a_{1(m+l)} \\ a_{21} & a_{22} & \cdot & \cdot & \cdot & a_{2(m+l)} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{n1} & a_{n2} & \cdot & \cdot & \cdot & a_{n(m+l)} \end{bmatrix} \odot \begin{bmatrix} a_{(n+1)1} & a_{(n+1)2} & \cdot & \cdot & \cdot & a_{(n+1)(m+l)} \\ a_{(n+2)1} & a_{(n+2)2} & \cdot & \cdot & \cdot & a_{(n+2)(m+l)} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{(n+t)1} & a_{(n+t)2} & \cdot & \cdot & \cdot & a_{(n+t)(m+l)} \end{bmatrix}^T,$$

$$\triangle_4\left(\prod(\mathscr{C})\right) = \begin{bmatrix} a_{(n+1)1} & a_{(n+1)2} & \cdot & \cdot & \cdot & a_{(n+1)(m+l)} \\ a_{(n+2)1} & a_{(n+2)2} & \cdot & \cdot & \cdot & a_{(n+2)(m+l)} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{(n+t)1} & a_{(n+t)2} & \cdot & \cdot & \cdot & a_{(n+t)(m+l)} \end{bmatrix} \odot \begin{bmatrix} a_{(n+1)1} & a_{(n+1)2} & \cdot & \cdot & \cdot & a_{(n+1)(m+l)} \\ a_{(n+2)1} & a_{(n+2)2} & \cdot & \cdot & \cdot & a_{(n+2)(m+l)} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{(n+t)1} & a_{(n+t)2} & \cdot & \cdot & \cdot & a_{(n+t)(m+l)} \end{bmatrix}^T.$$

Proof. By Definitions 2.4 and 3.1, we have $\prod(\mathscr{C})$ and $\prod(\mathscr{C}^{+})$ as follows:

$$\prod(\mathscr{C}) = M_{\mathscr{C}} \odot M_{\mathscr{C}}^{T} = \begin{bmatrix} a_{11} & a_{12} & \cdot & \cdot & \cdot & a_{1m} \\ a_{21} & a_{22} & \cdot & \cdot & \cdot & a_{2m} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{n1} & a_{n2} & \cdot & \cdot & \cdot & a_{nm} \end{bmatrix} \odot \begin{bmatrix} a_{11} & a_{12} & \cdot & \cdot & \cdot & a_{1m} \\ a_{21} & a_{22} & \cdot & \cdot & \cdot & a_{2m} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{n1} & a_{n2} & \cdot & \cdot & \cdot & a_{nm} \end{bmatrix}^{T}$$

$$= \begin{bmatrix} d_{11} & d_{12} & \cdot & \cdot & \cdot & d_{1n} \\ d_{21} & d_{22} & \cdot & \cdot & \cdot & d_{2n} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ d_{n1} & d_{n2} & \cdot & \cdot & \cdot & d_{nn} \end{bmatrix},$$

$$\prod(\mathscr{C}^{+}) = M_{\mathscr{C}^{+}} \odot M_{\mathscr{C}^{+}}^{T}$$

$$= \begin{bmatrix} a_{11} & a_{12} & \cdot & \cdot & \cdot & a_{1m} & a_{1(m+1)} & \cdot & \cdot & \cdot & a_{1(m+l)} \\ a_{21} & a_{22} & \cdot & \cdot & \cdot & a_{2m} & a_{2(m+1)} & \cdot & \cdot & \cdot & a_{2(m+l)} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{n1} & a_{n2} & \cdot & \cdot & \cdot & a_{nm} & a_{n(m+1)} & \cdot & \cdot & \cdot & a_{n(m+l)} \\ a_{(n+1)1} & a_{(n+1)2} & \cdot & \cdot & \cdot & a_{(n+1)m} & a_{(n+1)(m+1)} & \cdot & \cdot & \cdot & a_{(n+1)(m+l)} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{(n+t)1} & a_{(n+t)2} & \cdot & \cdot & \cdot & a_{(n+t)m} & a_{(n+t)(m+1)} & \cdot & \cdot & \cdot & a_{(n+t)(m+l)} \end{bmatrix} \odot$$

$$\begin{bmatrix} a_{11} & a_{12} & \cdot & \cdot & \cdot & a_{1m} & a_{1(m+1)} & \cdot & \cdot & \cdot & a_{1(m+l)} \\ a_{21} & a_{22} & \cdot & \cdot & \cdot & a_{2m} & a_{2(m+1)} & \cdot & \cdot & \cdot & a_{2(m+l)} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{n1} & a_{n2} & \cdot & \cdot & \cdot & a_{nm} & a_{n(m+1)} & \cdot & \cdot & \cdot & a_{n(m+l)} \\ a_{(n+1)1} & a_{(n+1)2} & \cdot & \cdot & \cdot & a_{(n+1)m} & a_{(n+1)(m+1)} & \cdot & \cdot & \cdot & a_{(n+1)(m+l)} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{(n+t)1} & a_{(n+t)2} & \cdot & \cdot & \cdot & a_{(n+t)m} & a_{(n+t)(m+1)} & \cdot & \cdot & \cdot & a_{(n+t)(m+l)} \end{bmatrix}^{T}$$

$$= \begin{bmatrix} e_{11} & e_{12} & \cdot & \cdot & \cdot & e_{1n} & e_{1(n+1)} & \cdot & \cdot & \cdot & e_{1(n+t)} \\ e_{21} & e_{22} & \cdot & \cdot & \cdot & e_{2n} & e_{2(n+1)} & \cdot & \cdot & \cdot & e_{2(n+t)} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ e_{n1} & e_{n2} & \cdot & \cdot & \cdot & e_{nn} & e_{n(n+1)} & \cdot & \cdot & \cdot & e_{n(n+t)} \\ e_{(n+1)1} & e_{(n+1)2} & \cdot & \cdot & \cdot & e_{(n+1)n} & e_{(n+1)(n+1)} & \cdot & \cdot & \cdot & e_{(n+1)(n+1)} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ e_{(n+t)1} & e_{(n+t)2} & \cdot & \cdot & \cdot & e_{(n+t)n} & e_{(n+t)(n+1)} & \cdot & \cdot & \cdot & e_{(n+t)(n+t)} \end{bmatrix}.$$

By Definition 2.4, we have that

$$\begin{aligned} e_{11} &= \begin{bmatrix} a_{11} & a_{12} & \cdot & \cdot & \cdot & a_{1m} & a_{1(m+1)} & \cdot & \cdot & \cdot & a_{1(m+l)} \end{bmatrix} \odot \\ &\quad \begin{bmatrix} a_{11} & a_{12} & \cdot & \cdot & \cdot & a_{1m} & a_{1(m+1)} & \cdot & \cdot & \cdot & a_{1(m+l)} \end{bmatrix}^{T} \\ &= \begin{bmatrix} a_{11} & a_{12} & \cdot & \cdot & \cdot & a_{1m} \end{bmatrix} \odot \begin{bmatrix} a_{11} & a_{12} & \cdot & \cdot & \cdot & a_{1m} \end{bmatrix}^{T} \wedge \\ &\quad \begin{bmatrix} a_{1(m+1)} & \cdot & \cdot & \cdot & a_{1(m+l)} \end{bmatrix} \odot \begin{bmatrix} a_{1(m+1)} & \cdot & \cdot & \cdot & a_{1(m+l)} \end{bmatrix}^{T} \\ &= d_{11} \wedge \begin{bmatrix} a_{1(m+1)} & \cdot & \cdot & \cdot & a_{1(m+l)} \end{bmatrix} \odot \begin{bmatrix} a_{1(m+1)} & \cdot & \cdot & \cdot & a_{1(m+l)} \end{bmatrix}^{T}, \\ e_{(n+1)1} &= \begin{bmatrix} a_{(n+1)1} & a_{(n+1)2} & \cdot & \cdot & \cdot & a_{(n+1)m} & a_{(n+1)(m+1)} & \cdot & \cdot & \cdot & a_{(n+1)(m+l)} \end{bmatrix} \odot \end{aligned}$$

$$\begin{aligned}
&\begin{bmatrix} a_{11} & a_{12} & \cdot & \cdot & \cdot & a_{1m} & a_{1(m+1)} & \cdot & \cdot & \cdot & a_{1(m+l)} \end{bmatrix}^T \\
&= 1 \wedge \begin{bmatrix} a_{(n+1)1} & a_{(n+1)2} & \cdot & \cdot & \cdot & a_{(n+1)m} & a_{(n+1)(m+1)} & \cdot & \cdot & \cdot & a_{(n+1)(m+l)} \end{bmatrix} \odot \\
&\begin{bmatrix} a_{11} & a_{12} & \cdot & \cdot & \cdot & a_{1m} & a_{1(m+1)} & \cdot & \cdot & \cdot & a_{1(m+l)} \end{bmatrix}^T, \\
e_{1(n+1)} &= \begin{bmatrix} a_{11} & a_{12} & \cdot & \cdot & \cdot & a_{1m} & a_{1(m+1)} & \cdot & \cdot & \cdot & a_{1(m+l)} \end{bmatrix} \odot \\
&\begin{bmatrix} a_{(n+1)1} & a_{(n+1)2} & \cdot & \cdot & \cdot & a_{(n+1)m} & a_{(n+1)(m+1)} & \cdot & \cdot & \cdot & a_{(n+1)(m+l)} \end{bmatrix}^T \\
&= 1 \wedge \begin{bmatrix} a_{11} & a_{12} & \cdot & \cdot & \cdot & a_{1m} & a_{1(m+1)} & \cdot & \cdot & \cdot & a_{1(m+l)} \end{bmatrix} \odot \\
&\begin{bmatrix} a_{(n+1)1} & a_{(n+1)2} & \cdot & \cdot & \cdot & a_{(n+1)m} & a_{(n+1)(m+1)} & \cdot & \cdot & \cdot & a_{(n+1)(m+l)} \end{bmatrix}^T, \\
e_{(n+1)(n+1)} &= \begin{bmatrix} a_{(n+1)1} & a_{(n+1)2} & \cdot & \cdot & \cdot & a_{(n+1)m} & a_{(n+1)(m+1)} & \cdot & \cdot & \cdot & a_{(n+1)(m+l)} \end{bmatrix} \odot \\
&\begin{bmatrix} a_{(n+1)1} & a_{(n+1)2} & \cdot & \cdot & \cdot & a_{(n+1)m} & a_{(n+1)(m+1)} & \cdot & \cdot & \cdot & a_{(n+1)(m+l)} \end{bmatrix}^T \\
&= 1 \wedge \begin{bmatrix} a_{(n+1)1} & a_{(n+1)2} & \cdot & \cdot & \cdot & a_{(n+1)m} & a_{(n+1)(m+1)} & \cdot & \cdot & \cdot & a_{(n+1)(m+l)} \end{bmatrix} \odot \\
&\begin{bmatrix} a_{(n+1)1} & a_{(n+1)2} & \cdot & \cdot & \cdot & a_{(n+1)m} & a_{(n+1)(m+1)} & \cdot & \cdot & \cdot & a_{(n+1)(m+l)} \end{bmatrix}^T.
\end{aligned}$$

Since $e_{11} \in \triangle_1(\prod(\mathscr{C}))$, $e_{(n+1)1} \in \triangle_2(\prod(\mathscr{C}))$, $e_{1(n+1)} \in \triangle_3(\prod(\mathscr{C}))$, and $e_{(n+1)(n+1)} \in \triangle_4(\prod(\mathscr{C}))$, we can compute the other elements of $\triangle_1(\prod(\mathscr{C}))$, $\triangle_2(\prod(\mathscr{C}))$, $\triangle_3(\prod(\mathscr{C}))$, and $\triangle_4(\prod(\mathscr{C}))$ similarly. To compute $\prod(\mathscr{C}^+)$ on the basis of $\prod(\mathscr{C})$, we only need to compute $\triangle_1(\prod(\mathscr{C}))$, $\triangle_2(\prod(\mathscr{C}))$, $\triangle_3(\prod(\mathscr{C}))$, and $\triangle_4(\prod(\mathscr{C}))$ as follows:

$$\triangle_1\left(\prod(\mathscr{C})\right) = \begin{bmatrix} a_{1(m+1)} & a_{2(m+1)} & \cdot & \cdot & \cdot & a_{n(m+1)} \\ a_{1(m+2)} & a_{2(m+2)} & \cdot & \cdot & \cdot & a_{n(m+2)} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{1(m+l)} & a_{2(m+l)} & \cdot & \cdot & \cdot & a_{n(m+l)} \end{bmatrix}^T \odot \begin{bmatrix} a_{1(m+1)} & a_{2(m+1)} & \cdot & \cdot & \cdot & a_{n(m+1)} \\ a_{1(m+2)} & a_{2(m+2)} & \cdot & \cdot & \cdot & a_{n(m+2)} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{1(m+l)} & a_{2(m+l)} & \cdot & \cdot & \cdot & a_{n(m+l)} \end{bmatrix},$$

$$\triangle_2\left(\prod(\mathscr{C})\right) = \begin{bmatrix} a_{(n+1)1} & a_{(n+1)2} & \cdot & \cdot & \cdot & a_{(n+1)(m+l)} \\ a_{(n+2)1} & a_{(n+2)2} & \cdot & \cdot & \cdot & a_{(n+2)(m+l)} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{(n+t)1} & a_{(n+t)2} & \cdot & \cdot & \cdot & a_{(n+t)(m+l)} \end{bmatrix} \odot \begin{bmatrix} a_{11} & a_{12} & \cdot & \cdot & \cdot & a_{1(m+l)} \\ a_{21} & a_{22} & \cdot & \cdot & \cdot & a_{2(m+l)} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{n1} & a_{n2} & \cdot & \cdot & \cdot & a_{n(m+l)} \end{bmatrix}^T,$$

$$\triangle_3\left(\prod(\mathscr{C})\right) = \begin{bmatrix} a_{11} & a_{12} & \cdot & \cdot & \cdot & a_{1(m+l)} \\ a_{21} & a_{22} & \cdot & \cdot & \cdot & a_{2(m+l)} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{n1} & a_{n2} & \cdot & \cdot & \cdot & a_{n(m+l)} \end{bmatrix} \odot \begin{bmatrix} a_{(n+1)1} & a_{(n+1)2} & \cdot & \cdot & \cdot & a_{(n+1)(m+l)} \\ a_{(n+2)1} & a_{(n+2)2} & \cdot & \cdot & \cdot & a_{(n+2)(m+l)} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{(n+t)1} & a_{(n+t)2} & \cdot & \cdot & \cdot & a_{(n+t)(m+l)} \end{bmatrix}^T,$$

$$\triangle_4\left(\prod(\mathscr{C})\right) = \begin{bmatrix} a_{(n+1)1} & a_{(n+1)2} & \cdot & \cdot & \cdot & a_{(n+1)(m+l)} \\ a_{(n+2)1} & a_{(n+2)2} & \cdot & \cdot & \cdot & a_{(n+2)(m+l)} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{(n+t)1} & a_{(n+t)2} & \cdot & \cdot & \cdot & a_{(n+t)(m+l)} \end{bmatrix} \odot \begin{bmatrix} a_{(n+1)1} & a_{(n+1)2} & \cdot & \cdot & \cdot & a_{(n+1)(m+l)} \\ a_{(n+2)1} & a_{(n+2)2} & \cdot & \cdot & \cdot & a_{(n+2)(m+l)} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{(n+t)1} & a_{(n+t)2} & \cdot & \cdot & \cdot & a_{(n+t)(m+l)} \end{bmatrix}^T.$$

Therefore, we have

$$\prod(\mathscr{C}^+) = \begin{bmatrix} \prod(\mathscr{C}) & 1 \\ 1 & 1 \end{bmatrix} \bigwedge \begin{bmatrix} \triangle_1(\prod(\mathscr{C})) & \triangle_3(\prod(\mathscr{C})) \\ \triangle_2(\prod(\mathscr{C})) & \triangle_4(\prod(\mathscr{C})) \end{bmatrix}.$$

□

Example 3.6 (Continued from Example 3.2). By Definition 2.4, we obtain that

$$\prod(\mathscr{C}) = M_{\mathscr{C}} \odot M_{\mathscr{C}}^T = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix} \odot \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}^T = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

By Theorem 3.5, we have that

$$\triangle_1\Big(\prod(\mathscr{C})\Big) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}^T \odot \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix},$$

$$\triangle_2\Big(\prod(\mathscr{C})\Big) = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \odot \begin{bmatrix} 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{bmatrix}^T = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\triangle_3\Big(\prod(\mathscr{C})\Big) = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{bmatrix}^T \odot \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}^T,$$

$$\triangle_4\Big(\prod(\mathscr{C})\Big) = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \odot \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}^T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Thus, we have

$$\prod(\mathscr{C}^+) = \begin{bmatrix} \prod(\mathscr{C}) & 1 \\ 1 & 1 \end{bmatrix} \bigwedge \begin{bmatrix} \triangle_1(\prod(\mathscr{C})) & \triangle_3(\prod(\mathscr{C})) \\ \triangle_2(\prod(\mathscr{C})) & \triangle_4(\prod(\mathscr{C})) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

By Definition 2.5, we obtain

$$\mathcal{X}_{XH(X)} = \prod(\mathscr{C}^+) \bullet \mathcal{X}_X = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}^T,$$

$$\mathcal{X}_{XL(X)} = \prod(\mathscr{C}^+) \odot \mathcal{X}_X = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \odot \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}^T.$$

Therefore, $XH(X) = \{x_2, x_3, x_4, x_5\}$ and $XL(X) = \{x_3, x_4, x_5\}$.

In Example 3.6, we must compute all of the elements in $\prod(\mathscr{C}^+)$ to construct the set approximations by Definition 2.4. In contrast, according to Theorem 3.5, we only need to calculate the elements in $\triangle_1(\prod(\mathscr{C}))$, $\triangle_2(\prod(\mathscr{C}))$, $\triangle_3(\prod(\mathscr{C}))$, and $\triangle_4(\prod(\mathscr{C}))$. Thus, the incremental approach is more effective for computing the approximations of sets.

In addition to the dynamic covering approximation spaces caused by the immigration of objects, practical situations also include dynamic covering approximation spaces caused by the emigration of objects.

Definition 3.7. Let $(U, \mathscr{C})$ and $(U^-, \mathscr{C}^-)$ be covering approximation spaces, where $U = \{x_1, x_2, \ldots, x_n\}$, $U^- = \{x_1, x_2, \ldots, x_{n-t}\} (t \geq 2)$, $\mathscr{C} = \{C_1, C_2, \ldots, C_m\}$, $\mathscr{C}^- = \{C_1^-, C_2^-, \ldots, C_{m-l}^-\} (l \geq 2)$, where $C_i^- = C_i - \Delta C_i$ $(1 \leq i \leq m-l)$, and $\Delta C_i \subseteq \{x_{n-t+1}, x_{n-t+2}, \ldots, x_n\}$. Then $(U^-, \mathscr{C}^-)$ is called a dynamic covering approximation space of $(U, \mathscr{C})$.

While some types of dynamic covering approximation spaces address the deletion of objects, we only discuss the type of dynamic covering approximation space shown in Definition 3.7 for simplicity.

Example 3.8. Let $(U, \mathscr{C})$ be a covering approximation space, where $U = \{x_1, x_2, x_3, x_4, x_5, x_6\}$, $\mathscr{C} = \{C_1, C_2, C_3, C_4, C_5\}$, $C_1 = \{x_1, x_4, x_5\}$, $C_2 = \{x_1, x_2, x_4, x_5\}$, $C_3 = \{x_3, x_4\}$, $C_4 = \{x_3, x_5, x_6\}$, and $C_5 = \{x_1, x_6\}$. By deleting $\{x_5, x_6\}$ from U, we get a dynamic covering approximation space $(U^-, \mathscr{C}^-)$ of $(U, \mathscr{C})$, where $U^- = \{x_1, x_2, x_3, x_4\}$, $\mathscr{C}^- = \{C_1^-, C_2^-, C_3^-\}$, $C_1^- = \{x_1, x_4\}$, $C_2^- = \{x_1, x_2, x_4\}$, and $C_3^- = \{x_3, x_4\}$.

We also show how to construct $\Gamma(\mathscr{C}^-)$ based on $\Gamma(\mathscr{C})$. For convenience, we denote $M_{\mathscr{C}} = (a_{ij})_{n\times m}$, $M_{\mathscr{C}^-} = (a_{ij})_{(n-t)\times(m-l)}$, $\Gamma(\mathscr{C}) = (b_{ij})_{n\times n}$, and $\Gamma(\mathscr{C}^-) = (c_{ij}^-)_{(n-t)\times(n-t)}$.

Theorem 3.9. *Let $(U^-, \mathscr{C}^-)$ be a dynamic covering approximation space of $(U, \mathscr{C})$, $\Gamma(\mathscr{C})$ and $\Gamma(\mathscr{C}^-)$ the type-1 characteristic matrices of $\mathscr{C}$ and $\mathscr{C}^-$, respectively. Then*

$$c_{ij}^- = \begin{cases} 0, & b_{ij} = 0; \\ 1, & b_{ij} = 1 \wedge \triangle c_{ij} = 0; \\ \begin{bmatrix} a_{i1} & a_{i2} & . & . & . & a_{i(m-l)} \end{bmatrix} \bullet \begin{bmatrix} a_{j1} & a_{j2} & . & . & . & a_{j(m-l)} \end{bmatrix}^T, & b_{ij} = 1 \wedge \triangle c_{ij} = 1. \end{cases}$$

where

$$\triangle c_{ij} = \begin{bmatrix} a_{i(m-l+1)} & a_{i(m-l+2)} & . & . & . & a_{im} \end{bmatrix} \bullet \begin{bmatrix} a_{j(m-l+1)} & a_{j(m-l+2)} & . & . & . & a_{jm} \end{bmatrix}^T.$$

Proof. It is straightforward by Theorem 3.3. □

Example 3.10 (Continued from Example 3.8)**.** Taking $X = \{x_3, x_4\}$. By Definition 2.4, we first have that

$$\Gamma(\mathscr{C}) = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}.$$

Second, by Theorem 3.9, we get that

$$\triangle_1(\Gamma(\mathscr{C}^-)) = (\triangle c_{ij})_{4\times 4} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}^T \bullet \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

and

$$\Gamma(\mathscr{C}^-) = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}.$$

By Definition 2.5, we have that

$$\mathcal{X}_{SH(X)} = \Gamma(\mathscr{C}^-) \bullet \mathcal{X}_X = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \bullet \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}^T,$$

$$\mathcal{X}_{SL(X)} = \Gamma(\mathscr{C}^-) \odot \mathcal{X}_X = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \odot \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}^T.$$

Therefore, $SH(X) = \{x_1, x_2, x_3, x_4\}$ and $SL(X) = \{x_3\}$.

We also show how to construct $\prod(\mathscr{C}^-)$ based on $\prod(\mathscr{C})$. For convenience, we denote $M_{\mathscr{C}} = (a_{ij})_{n\times m}$, $M_{\mathscr{C}^-} = (a_{ij})_{(n-t)\times(m-l)}$, $\prod(\mathscr{C}) = (d_{ij})_{n\times n}$, and $\prod(\mathscr{C}^-) = (e_{ij}^-)_{(n-t)\times(n-t)}$.

Theorem 3.11. *Let $(U^-, \mathscr{C}^-)$ be a dynamic covering approximation space of $(U, \mathscr{C})$, $\prod(\mathscr{C})$ and $\prod(\mathscr{C}^-)$ the type-2 characteristic matrices of $\mathscr{C}$ and $\mathscr{C}^-$, respectively. Then*

$$e_{ij}^- = \begin{cases} 1, & d_{ij} = 1 \wedge \triangle e_{ij} = 1; \\ 0, & d_{ij} = 0 \wedge \triangle e_{ij} = 1; \\ \begin{bmatrix} a_{i1} & a_{i2} & . & . & . & a_{i(m-l)} \end{bmatrix} \odot \begin{bmatrix} a_{j1} & a_{j2} & . & . & . & a_{j(m-l)} \end{bmatrix}^T, & d_{ij} = 0 \wedge \triangle e_{ij} = 0. \end{cases}$$

where

$$\triangle e_{ij} = \begin{bmatrix} a_{i(m-l+1)} & a_{i(m-l+2)} & . & . & . & a_{im} \end{bmatrix} \odot \begin{bmatrix} a_{j(m-l+1)} & a_{j(m-l+2)} & . & . & . & a_{jm} \end{bmatrix}^T.$$

Proof. It is straightforward by Theorem 3.5. □

Example 3.12 (Continued from Example 3.10)**.** By Definition 2.4, we first have that

$$\prod(\mathscr{C}) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Second, by Theorem 3.11, we get that

$$\triangle_1\left(\prod(\mathscr{C}^-)\right) = (\triangle e_{ij})_{4\times4} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}^T \odot \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

and

$$\prod(\mathscr{C}^-) = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

By Definition 2.5, we obtain

$$\mathcal{X}_{XH(X)} = \prod(\mathscr{C}^-) \bullet \mathcal{X}_X = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}^T,$$

$$\mathcal{X}_{XL(X)} = \prod(\mathscr{C}^-) \odot \mathcal{X}_X = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \odot \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 1 \end{bmatrix}^T.$$

Therefore, $XH(X) = \{x_1, x_2, x_3, x_4\}$ and $XL(X) = \{x_3, x_4\}$.

Remark. In fact, if some objects are added to the covering approximation space as others are leaving the covering approximation space, then the type-1 and type-2 characteristic matrices of dynamic coverings can be computed using two steps as follows: (1) compute the type-1 and type-2 characteristic matrices of dynamic coverings by Theorems 3.3 and 3.5, respectively; (2) construct the type-1 and type-2 characteristic matrices of dynamic coverings by Theorems 3.9 and 3.11, respectively. In our big data era, practical applications use more dynamic covering approximation spaces corresponding to Definition 3.1 than Definition 3.7. Therefore, the following discussion focuses on the dynamic covering approximation spaces given by Definition 3.1.

4. Non-incremental and incremental algorithms for computing the second and sixth lower and upper approximations of sets

In this section, we show non-incremental and incremental algorithms of computing the second lower and upper approximations of sets.

Algorithm 4.1. (Non-incremental algorithm of computing $SH_{\mathscr{C}^+}(X^+)$ and $SL_{\mathscr{C}^+}(X^+)$(**NIS**))
Step 1: Input $(U^+, \mathscr{C}^+)$ and $X^+ \subseteq U^+$;
Step 2: Construct $M_{\mathscr{C}^+}$ and $\Gamma(\mathscr{C}^+) = M_{\mathscr{C}^+} \bullet M^T_{\mathscr{C}^+}$;
Step 3: Compute $\mathcal{X}_{SH_{\mathscr{C}^+}(X^+)} = \Gamma(\mathscr{C}^+) \bullet \mathcal{X}_{X^+}$ and $\mathcal{X}_{SL_{\mathscr{C}^+}(X^+)} = \Gamma(\mathscr{C}^+) \odot \mathcal{X}_{X^+}$;
Step 4: Output $SH_{\mathscr{C}^+}(X^+)$ and $SL_{\mathscr{C}^+}(X^+)$.

Algorithm 4.2. (Incremental algorithm of computing $SH_{\mathscr{C}^+}(X^+)$ and $SL_{\mathscr{C}^+}(X^+)$(**IS**))
Step 1: Input $(U, \mathscr{C})$, $(U^+, \mathscr{C}^+)$, and $X \subseteq U^+$;
Step 2: Calculate $\Gamma(\mathscr{C}) = M_{\mathscr{C}} \bullet M^T_{\mathscr{C}}$, where $M_{\mathscr{C}} = (a_{ij})_{n\times m}$;
Step 3: Compute $\triangle_1(\Gamma(\mathscr{C}))$, $\triangle_2(\Gamma(\mathscr{C}))$, and $\triangle_3(\Gamma(\mathscr{C}))$;
Step 4: Construct $\Gamma(\mathscr{C}^+) = \begin{bmatrix} \Gamma(\mathscr{C}) & 0 \\ 0 & 0 \end{bmatrix} \vee \begin{bmatrix} \triangle_1(\Gamma(\mathscr{C})) & (\triangle_2(\Gamma(\mathscr{C})))^T \\ \triangle_2(\Gamma(\mathscr{C})) & \triangle_3(\Gamma(\mathscr{C})) \end{bmatrix}$;
Step 5: Obtain $\mathcal{X}_{SH(X)} = \Gamma(\mathscr{C}^+) \bullet \mathcal{X}_X$ and $\mathcal{X}_{SL(X)} = \Gamma(\mathscr{C}^+) \odot \mathcal{X}_X$;
Step 6: Output $SH_{\mathscr{C}^+}(X^+)$ and $SL_{\mathscr{C}^+}(X^+)$.

The time complexity of computing the second lower and upper approximations of sets is $O((m+l)\cdot(n^2+2nt+t^2+n))$ using Algorithm 4.1, and $O((m+l)\cdot(2nt+t^2+n))$ is the time complexity of Algorithm 4.2. Therefore, the time complexity of the incremental algorithm is lower than that of the non-incremental algorithm.

Table 1
Dynamic covering approximation spaces for experiments.

No.	Name	$\lvert U_i\rvert$	$\lvert\mathscr{C}_i\rvert$	$\lvert U_i^+\rvert$	$\lvert\mathscr{C}_i^+\rvert$
1	$(U_1, \mathscr{C}_1)$	3000	300	3100	330
2	$(U_2, \mathscr{C}_2)$	6000	600	6200	660
3	$(U_3, \mathscr{C}_3)$	9000	900	9300	990
4	$(U_4, \mathscr{C}_4)$	12,000	1200	12,400	1320
5	$(U_5, \mathscr{C}_5)$	15,000	1500	15,500	1650
6	$(U_6, \mathscr{C}_6)$	18,000	1800	18,600	1980
7	$(U_7, \mathscr{C}_7)$	21,000	2100	21,700	2310
8	$(U_8, \mathscr{C}_8)$	24,000	2400	24,800	2640
9	$(U_9, \mathscr{C}_9)$	27,000	2700	27,900	2970
10	$(U_{10}, \mathscr{C}_{10})$	30,000	3000	31,000	3300

Table 2
Computational times using Algorithms 4.1–4.4 in $(U_1^+, \mathscr{C}_1^+)$.

$t(s)$	1	2	3	4	5	6	7	8	9	10	$\bar{t}$
NIS	0.5766	0.5528	0.5703	0.5663	0.5884	0.5434	0.5610	0.5652	0.5391	0.5923	0.5656
NIX	1.0647	1.0600	1.0144	1.0132	1.0120	1.0119	1.0121	1.0189	1.0161	1.0196	1.0243
IS	0.2279	0.2259	0.2247	0.2262	0.2258	0.2255	0.2248	0.2267	0.2263	0.2259	0.2260
IX	0.4790	0.4713	0.4719	0.4721	0.4718	0.4717	0.4698	0.4700	0.4693	0.4698	0.4717

Subsequently, we present non-incremental and incremental algorithms of computing the sixth lower and upper approximations of sets.

Algorithm 4.3. (Non-incremental algorithm of computing $XH_{\mathscr{C}^+}(X^+)$ and $XL_{\mathscr{C}^+}(X^+)$(**NIX**))
Step 1: Input $(U^+, \mathscr{C}^+)$ and $X^+ \subseteq U^+$;
Step 2: Construct $M_{\mathscr{C}^+}$ and $\prod(\mathscr{C}^+) = M_{\mathscr{C}^+} \bullet M^T_{\mathscr{C}^+}$;
Step 3: Compute $\mathcal{X}_{XH_{\mathscr{C}^+}(X^+)} = \prod(\mathscr{C}^+) \bullet \mathcal{X}_{X^+}$ and $\mathcal{X}_{XL_{\mathscr{C}^+}(X^+)} = \prod(\mathscr{C}^+) \odot \mathcal{X}_{X^+}$;
Step 4: Output $XH_{\mathscr{C}^+}(X^+)$ and $XL_{\mathscr{C}^+}(X^+)$.

Algorithm 4.4. (Incremental algorithm of computing $XH_{\mathscr{C}^+}(X^+)$ and $XL_{\mathscr{C}^+}(X^+)$(**IX**))
Step 1: Input $(U, \mathscr{C})$, $(U^+, \mathscr{C}^+)$, and $X \subseteq U^+$;
Step 2: Construct $\prod(\mathscr{C})$, where $\prod(\mathscr{C}) = M_{\mathscr{C}} \odot M^T_{\mathscr{C}}$;
Step 3: Compute $\triangle_1(\prod(\mathscr{C}))$, $\triangle_2(\prod(\mathscr{C}))$, $\triangle_3(\prod(\mathscr{C}))$, and $\triangle_4(\prod(\mathscr{C}))$;
Step 4: Calculate $\prod(\mathscr{C}^+)$, where $\prod(\mathscr{C}^+) = \begin{bmatrix} \prod(\mathscr{C}) & 1 \\ 1 & 1 \end{bmatrix} \bigwedge \begin{bmatrix} \triangle_1(\prod(\mathscr{C})) & \triangle_3(\prod(\mathscr{C})) \\ \triangle_2(\prod(\mathscr{C})) & \triangle_4(\prod(\mathscr{C})) \end{bmatrix}$;
Step 5: Get $XH_{\mathscr{C}^+}(X^+) = \prod(\mathscr{C}^+) \bullet \mathcal{X}_X$ and $XL_{\mathscr{C}^+}(X^+) = \prod(\mathscr{C}^+) \odot \mathcal{X}_X$;
Step 6: Output $XH_{\mathscr{C}^+}(X^+)$ and $XL_{\mathscr{C}^+}(X^+)$.

The time complexity of computing the sixth lower and upper approximations of sets is $O((m+l) \cdot (n^2 + 2nt + t^2 + n))$ by Algorithm 4.3, and $O((m+l) \cdot (2nt + t^2 + n))$ is the time complexity of Algorithm 4.4. Therefore, the time complexity of the incremental algorithm is lower than that of the non-incremental algorithm.

5. Experimental analysis

In this section, we perform experiments to illustrate the effectiveness of Algorithms 4.1–4.4 for computing the second and sixth lower and upper approximations of concepts in dynamic covering approximation spaces when the covering cardinalities increase with object immigration.

To test Algorithms 4.1–4.4, we generated the 10 artificial covering approximation spaces $\{(U_i, \mathscr{C}_i) | i = 1, 2, 3, \ldots, 10\}$ outlined in Table 1, where $|U_i|$ indicates the number of objects in U_i, $|\mathscr{C}_i|$ denotes the cardinality of $\mathscr{C}_i$, $|U_i^+|$ is the number of objects in U_i^+, and $|\mathscr{C}_i^+|$ stands for the cardinality of $\mathscr{C}_i^+$. We conducted all computations on a PC with an Intel(R) Dual-Core(TM) i5-4590 CPU @ 3.30GHZ and 8GB memory, running 64-bit Windows 7; we used 64-bit Matlab R2009b as the software.

Here, we discuss the performance of Algorithms 4.1–4.4 in the dynamic covering approximation space $(U_i^+, \mathscr{C}_i^+)$ $(1 \leq i \leq 10)$. For example, we conduct the experiment on Algorithms 4.1–4.4 in the dynamic covering approximation space $(U_1^+, \mathscr{C}_1^+)$ as follows. First, we generate the covering approximation space $(U_1, \mathscr{C}_1)$ and compute $\Gamma(\mathscr{C}_1)$ and $\prod(\mathscr{C}_1)$ by Definition 2.4. Second, we get the dynamic covering approximation space $(U_1^+, \mathscr{C}_1^+)$ by randomly adding 100 objects into U_1 randomly and then construct $SH_{\mathscr{C}^+}(X^+)$, $SL_{\mathscr{C}^+}(X^+)$, $XH_{\mathscr{C}^+}(X^+)$, and $XL_{\mathscr{C}^+}(X^+)$ for any $X^+ \subseteq U^+$ using Algorithms 4.1 and 4.3. Third, we compute $SH_{\mathscr{C}^+}(X^+)$, $SL_{\mathscr{C}^+}(X^+)$, $XH_{\mathscr{C}^+}(X^+)$, and $XL_{\mathscr{C}^+}(X^+)$ with $\Gamma(\mathscr{C}_1)$ and $\prod(\mathscr{C}_1)$ using Algorithms 4.2 and 4.4. More details on the efficiency of Algorithms 4.1–4.4 for computing the second and sixth lower and upper approximations of concepts in dynamic covering approximation spaces $(U_1^+, \mathscr{C}_1^+)$ are shown in Table 2 and Fig. 1.

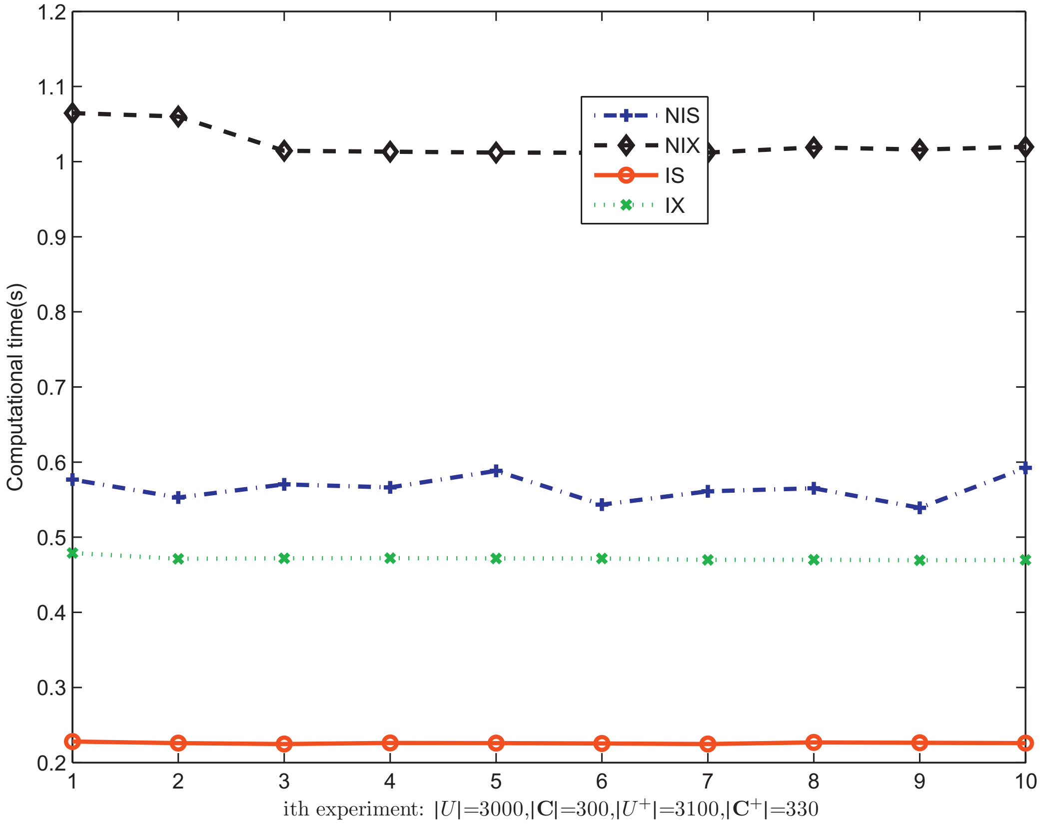

Fig. 1. Computational times using Algorithms 4.1–4.4 in $(U_1^+, \mathscr{C}_1^+)$.

Table 3
Computational times using Algorithms 4.1–4.4 in $(U_2^+, \mathscr{C}_2^+)$.

$t(s)$	1	2	3	4	5	6	7	8	9	10	$\bar{t}$
NIS	2.3931	2.3818	2.3230	2.3694	2.3851	2.4041	2.4308	2.2320	2.4373	2.3127	2.3669
NIX	4.4482	4.2829	4.2735	4.2780	4.2734	4.2806	4.2866	4.2900	4.2822	4.2677	4.2963
IS	0.8993	0.8950	0.8986	0.8994	0.8996	0.9002	0.8951	0.8954	0.8959	0.8978	0.8976
IX	1.9910	2.0004	2.0160	1.9953	1.9918	1.9921	1.9905	1.9965	1.9935	1.9926	1.9960

Table 4
Computational times using Algorithms 4.1–4.4 in $(U_3^+, \mathscr{C}_3^+)$.

$t(s)$	1	2	3	4	5	6	7	8	9	10	$\bar{t}$
NIS	5.5909	5.5188	5.5172	5.3116	5.3532	5.3160	5.1746	5.6186	5.5001	5.3365	5.4238
NIX	10.2217	9.8359	9.8936	9.8488	9.8771	9.8565	9.8427	9.8391	9.8508	9.8477	9.8914
IS	2.0888	2.0884	2.0763	2.0830	2.0873	2.0859	2.0855	2.0900	2.0909	2.0840	2.0860
IX	4.6341	4.6266	4.6279	4.6265	4.6193	4.6362	4.6270	4.6146	4.6247	4.6287	4.6266

Table 5
Computational times using Algorithms 4.1–4.4 in $(U_4^+, \mathscr{C}_4^+)$.

$t(s)$	1	2	3	4	5	6	7	8	9	10	$\bar{t}$
NIS	9.9489	9.5870	9.3932	9.7759	9.8139	9.3773	9.9289	9.6748	9.6136	9.7154	9.6829
NIX	18.3277	17.7100	17.7250	18.1853	17.7506	17.7024	17.7359	17.6979	17.6949	17.7210	17.8251
IS	3.7438	3.7264	3.7357	3.7192	3.7425	3.7280	3.7196	3.7389	3.7171	3.7524	3.7323
IX	8.2458	8.2727	8.2466	8.2651	8.2527	8.2402	8.2461	8.2497	8.2482	8.2539	8.2521

Table 6
Computational times using Algorithms 4.1–4.4 in $(U_5^+, \mathscr{C}_5^+)$.

$t(s)$	1	2	3	4	5	6	7	8	9	10	$\bar{t}$
NIS	15.4719	15.7468	15.3756	15.5304	15.7810	15.6241	15.8075	15.5052	15.5259	15.5355	15.5904
NIX	29.1151	28.2757	28.2586	28.3687	28.2356	28.2540	28.2598	28.2242	28.2045	28.1906	28.3387
IS	5.9551	5.9533	5.8691	5.9108	5.9617	5.8954	5.9277	5.9030	5.9424	5.9095	5.9228
IX	12.9988	12.9884	13.0431	12.9852	12.9986	12.9992	13.0020	13.0177	13.0253	13.0030	13.0061

Table 7
Computational times using Algorithms 4.1–4.4 in $(U_6^+, \mathscr{C}_6^+)$.

$t(s)$	1	2	3	4	5	6	7	8	9	10	$\bar{t}$
NIS	22.8132	22.4039	23.2349	22.7676	23.0889	22.8922	22.9186	22.3781	23.0388	22.6357	22.8172
NIX	44.2780	42.0411	42.1306	42.0110	42.2207	43.0541	42.0884	42.5397	42.7606	42.0623	42.5187
IS	8.5779	8.5281	8.5149	8.5486	8.5169	8.5231	8.5622	8.5476	8.5361	8.5404	8.5396
IX	19.0485	18.9816	19.0388	19.0208	19.0496	18.9860	19.0062	19.0007	18.9840	19.0248	19.0141

Table 8
Computational times using Algorithms 4.1–4.4 in $(U_7^+, \mathscr{C}_7^+)$.

$t(s)$	1	2	3	4	5	6	7	8	9	10	$\bar{t}$
NIS	31.0927	31.1847	30.8458	31.3568	31.5426	30.9888	31.1460	31.2257	31.1842	31.5378	31.2105
NIX	59.2375	57.3649	57.2514	57.2193	57.3211	57.5383	57.1850	57.3001	57.4061	57.3427	57.5166
IS	11.7289	11.6272	11.5839	11.5784	11.5785	11.5997	11.6423	11.6269	11.6137	11.6315	11.6211
IX	25.6920	25.7110	25.6686	25.6755	25.6094	25.5876	25.6706	25.6168	25.5869	25.6433	25.6462

Table 9
Computational times using Algorithms 4.1–4.4 in $(U_8^+, \mathscr{C}_8^+)$.

$t(s)$	1	2	3	4	5	6	7	8	9	10	$\bar{t}$
NIS	41.5778	41.7299	41.6913	41.6440	41.3423	41.7533	42.4163	41.1085	41.6271	41.3819	41.6273
NIX	80.4870	79.0390	79.4425	78.4537	79.9899	78.4161	79.4013	81.2003	79.1413	78.6763	79.4247
IS	15.3122	15.3201	15.1341	15.1039	15.1491	15.2782	15.2727	15.1680	15.1152	15.2665	15.2120
IX	34.0190	33.8273	33.7320	33.7929	33.8485	33.8978	33.9340	33.7013	33.8686	33.7953	33.8417

Table 10
Computational times using Algorithms 4.1–4.4 in $(U_9^+, \mathscr{C}_9^+)$.

$t(s)$	1	2	3	4	5	6	7	8	9	10	$\bar{t}$
NIS	54.7912	54.3813	54.9239	53.8932	55.0448	53.9784	54.0926	55.4141	54.1229	54.6915	54.5334
NIX	110.8912	109.0519	109.1868	108.2111	110.4700	109.2189	109.4680	107.5752	108.7918	109.5043	109.2369
IS	20.0906	19.8863	19.7355	19.7147	19.7915	19.8553	19.6959	19.7304	19.9268	19.8015	19.8229
IX	44.8359	45.4595	44.9656	44.9720	44.6773	44.3519	44.6125	45.1804	45.0603	44.9016	44.9017

Table 11
Computational times using Algorithms 4.1–4.4 in $(U_{10}^+, \mathscr{C}_{10}^+)$.

$t(s)$	1	2	3	4	5	6	7	8	9	10	$\bar{t}$
NIS	66.9402	68.2013	66.6829	67.3929	67.7175	68.3884	67.2254	68.2567	67.8276	68.4374	67.7070
NIX	141.4040	143.5680	140.3704	143.4606	141.1702	143.1844	142.4325	142.6242	139.9998	141.3126	141.9527
IS	26.7875	27.0357	26.7383	27.0259	27.0075	26.6698	26.7841	26.7168	26.7118	27.0667	26.8544
IX	60.1791	60.9036	61.2077	60.5229	60.0135	62.2785	61.9972	60.7017	60.1402	60.4516	60.8396

To confirm the efficiency of Algorithms 4.2 and 4.4, the other experimental results are listed in Tables 3–11 and illustrated in Figs. 2–10 for dynamic covering approximation spaces $(U_i^+, \mathscr{C}_i^+)$ $(2 \leq i \leq 10)$. The experimental results show that Algorithms 4.2 and 4.4 execute faster than Algorithms 4.1 and 4.3, respectively, in dynamic covering approximation spaces. Therefore, Algorithms 4.2 and 4.4 are more efficient for computing the second and sixth lower and upper approximations of sets, respectively, in dynamic covering approximation spaces.

Fig. 11 shows the average execution times to compute the second and sixth lower and upper approximations of concepts in $(U_i^+, \mathscr{C}_i^+)$ $(1 \leq i \leq 10)$. We see that the computational times of Algorithms 4.1–4.4 usually increase as the number of objects and cardinalities of coverings increase. However, we find that Algorithms 4.2 and 4.4 are much faster than Algorithms 4.1 and 4.3, respectively, for computing the approximations of concepts when adding more objects. Therefore, Algorithms 4.2 and 4.4 are more efficient for computing the second and sixth lower and upper approximations of sets, respectively, in dynamic covering approximation spaces with large numbers of objects and cardinalities of coverings.

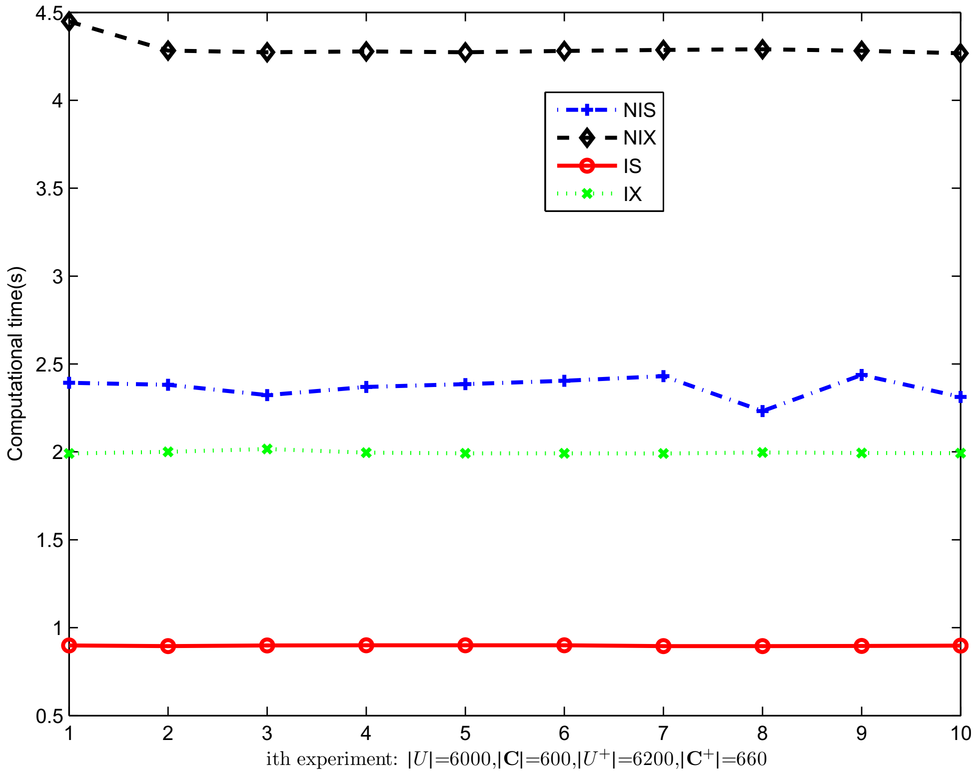

Fig. 2. Computational times using Algorithms 4.1–4.4 in $(U_2^+, \mathscr{C}_2^+)$.

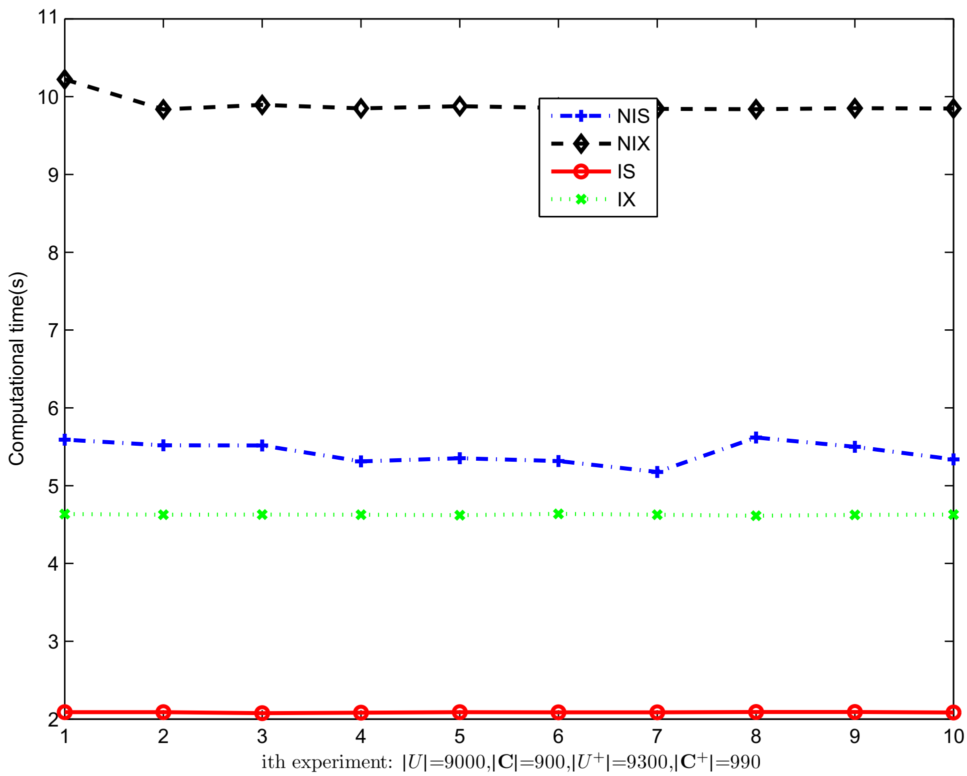

Fig. 3. Computational times using Algorithms 4.1–4.4 in $(U_3^+, \mathscr{C}_3^+)$.

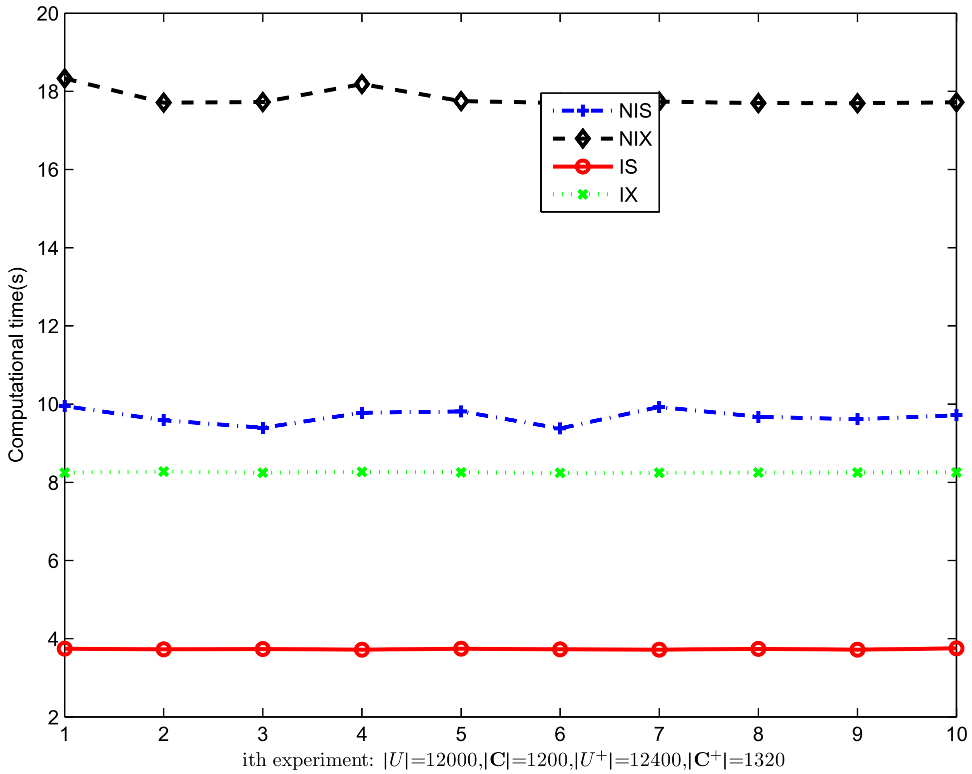

Fig. 4. Computational times using Algorithms 4.1–4.4 in $(U_4^+, \mathscr{C}_4^+)$.

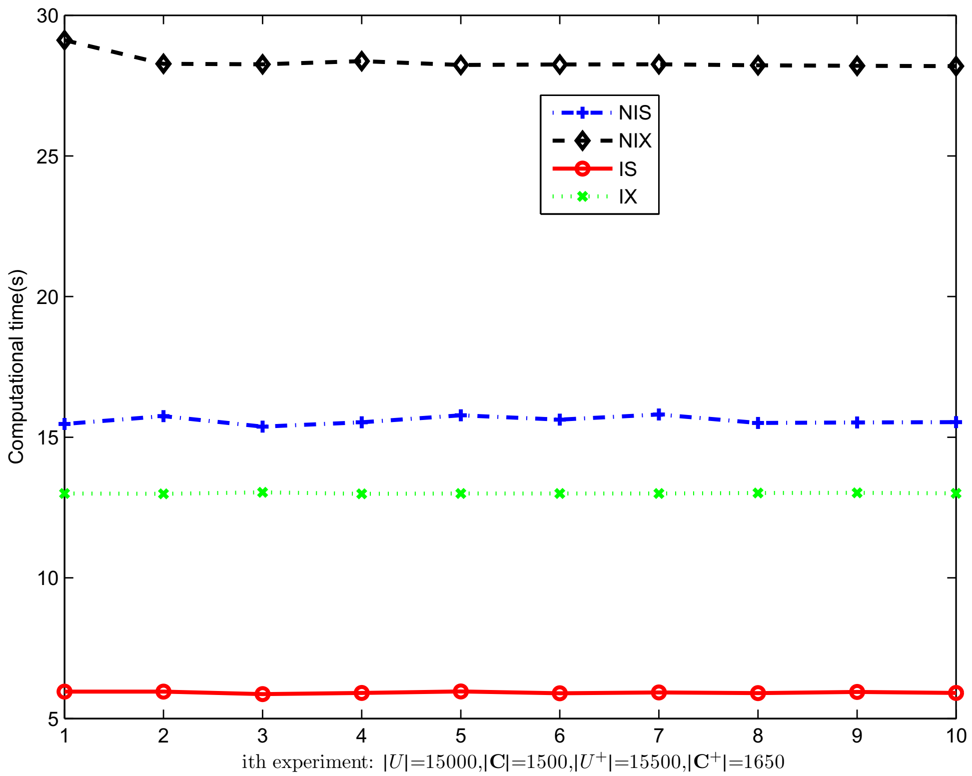

Fig. 5. Computational times using Algorithms 4.1–4.4 in $(U_5^+, \mathscr{C}_5^+)$.

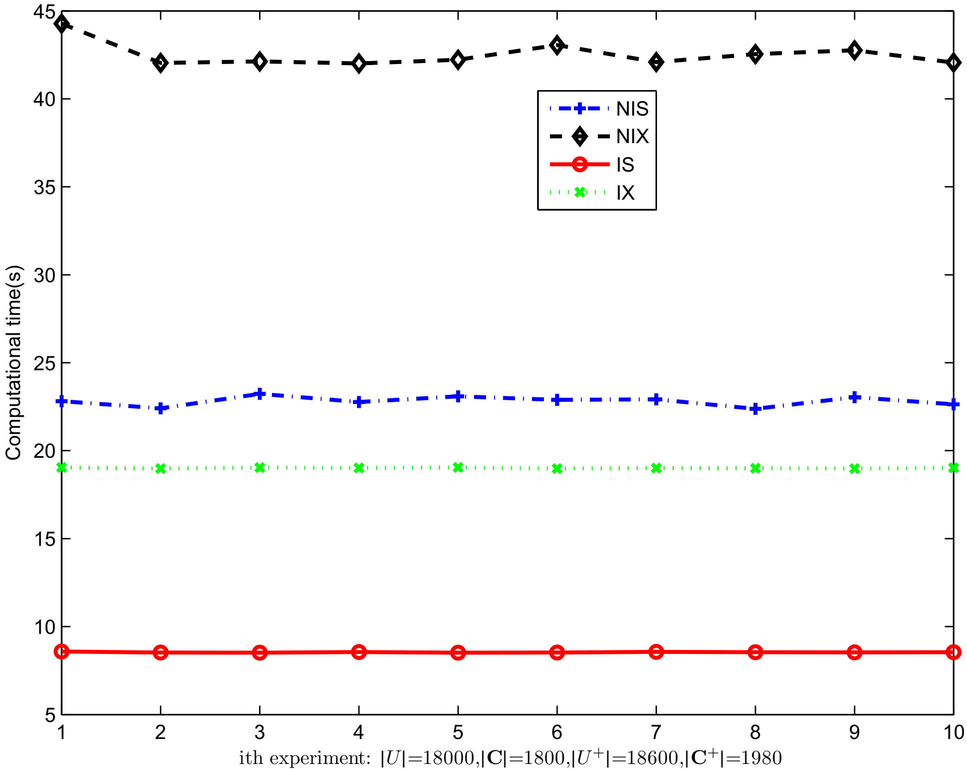

Fig. 6. Computational times using Algorithms 4.1–4.4 in $(U_6^+, \mathscr{C}_6^+)$.

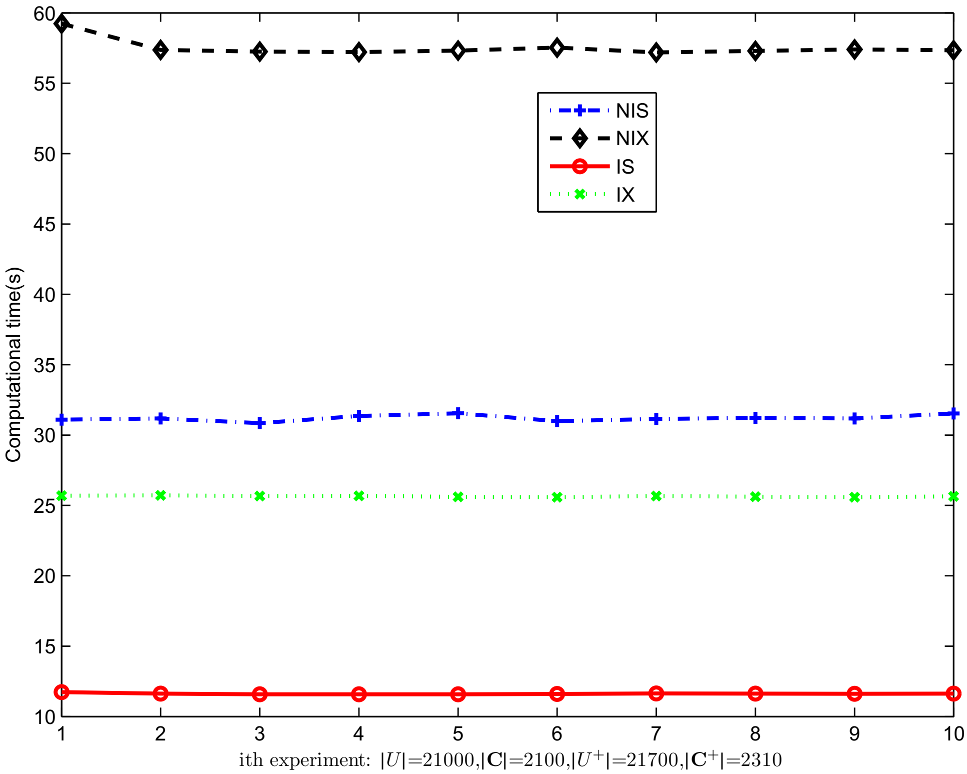

Fig. 7. Computational times using Algorithms 4.1–4.4 in $(U_7^+, \mathscr{C}_7^+)$.

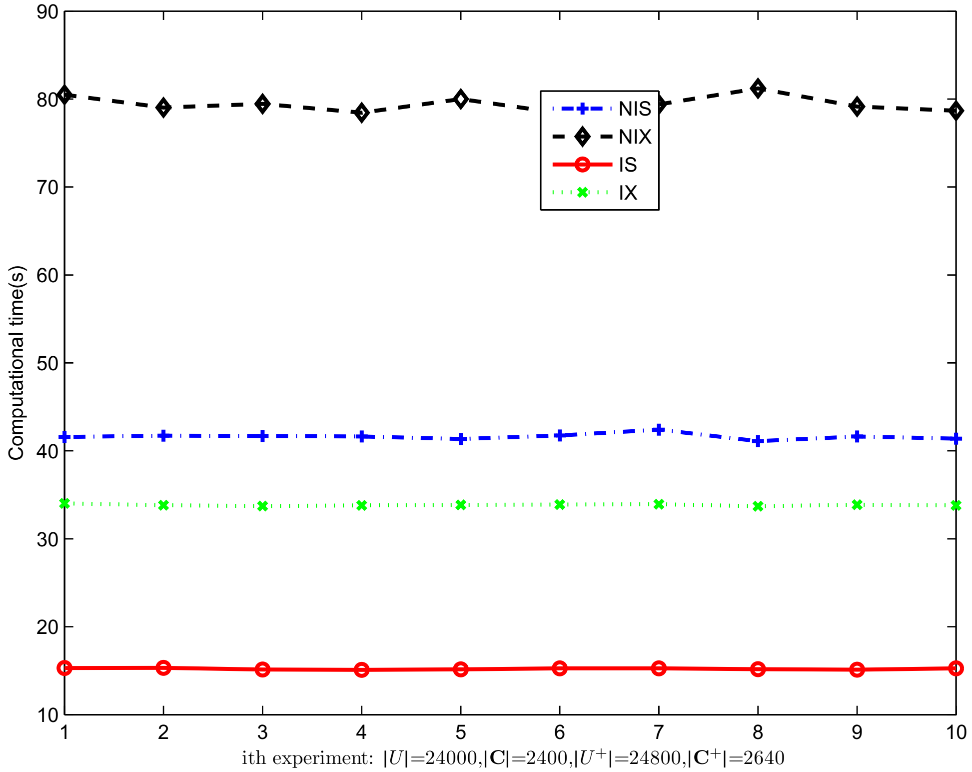

Fig. 8. Computational times using Algorithms 4.1–4.4 in $(U_8^+, \mathscr{C}_8^+)$.

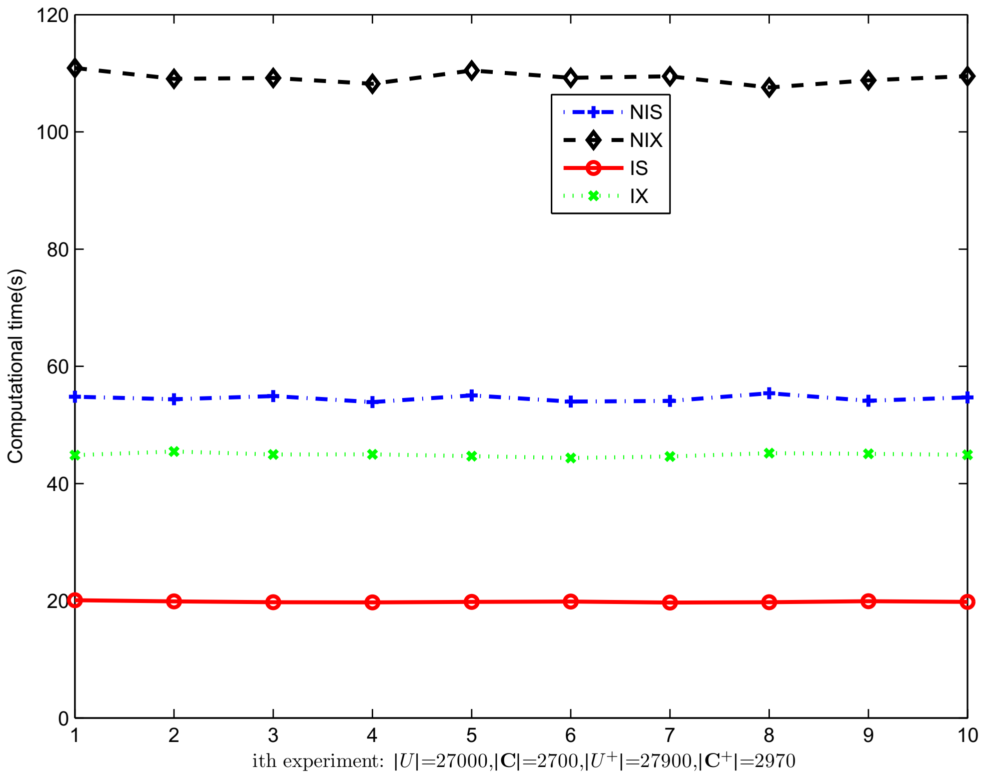

Fig. 9. Computational times using Algorithms 4.1–4.4 in $(U_9^+, \mathscr{C}_9^+)$.

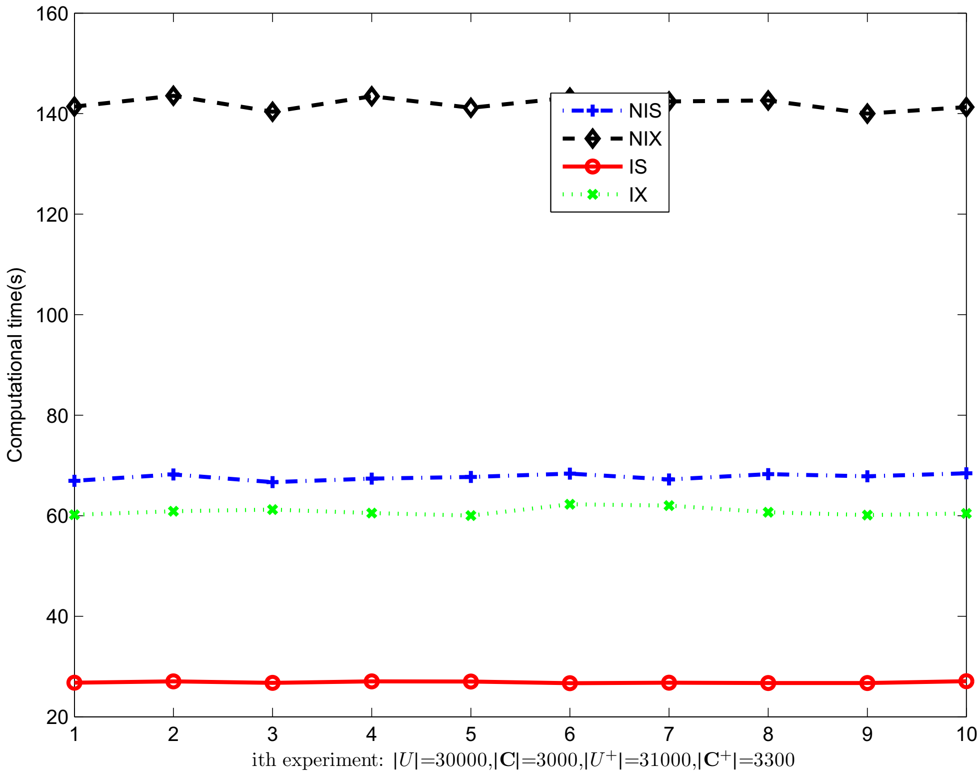

Fig. 10. Computational times using Algorithms 4.1–4.4 in $(U^+_{10}, \mathscr{C}^+_{10})$.

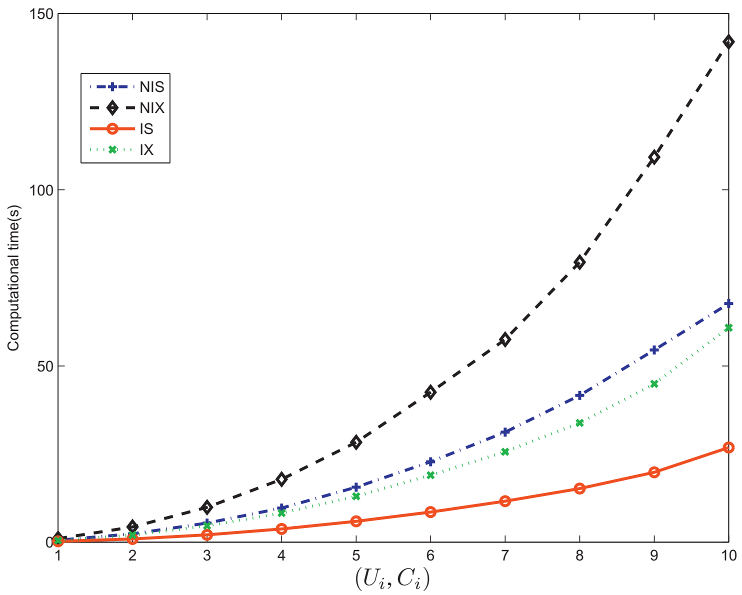

Fig. 11. Computational times using Algorithms 4.1–4.4 in $(U^+_i, \mathscr{C}^+_i)$, where $i = 1, 2, 3, \ldots, 10$.

In Tables 2–11, $t(s)$ denotes that the measure of time is in seconds; $\bar{t}$ indicates the average execution time of the corresponding algorithm in seconds; *NIS*, *IS*, *NIX*, and *IX* mean Algorithms 4.1–4.4, respectively; i stands for the experiment number on the x-axes in Figs. 1–10, and i denotes $(U_i, \mathscr{C}_i)$ on the x-axes in Fig. 11, while the y-axes correspond to the computational time to construct the approximations of concepts in dynamic covering approximation spaces.

Remark. In the experiment, we transform datasets downloaded from the University of California at Irvine (UCI)s repository of machine learning databases[10] into covering approximation spaces. Following the approach of Ref. [17], we transformed the Balance Scale Weight and Distance Database with four conditional attributes into the covering approximation space $(U,\mathscr{C})$, where $|U|=625$ and $|\mathscr{C}|=20$. Because there are five attribute values for each conditional attribute, we obtained a covering with five elements for each conditional attribute. Subsequently, based on Left-Weight, Left-Distance, Right-Weight, and Right-Distance, we obtained the covering approximation space $(U,\mathscr{C})$, where $|U|=625$ and $|\mathscr{C}|=20$. Because the purpose of the experiment is to test the efficiency of Algorithms 4.1–4.4 and the transformation process is more time-consuming than the computation, we generated 10 artificial covering approximation spaces $\{(U_i,\mathscr{C}_i)|i=1,2,3,\ldots,10\}$ for our experiments.

6. Knowledge reduction of dynamic covering decision information systems when increasing covering cardinalities

In this section, we perform knowledge reduction of dynamic covering decision information systems when covering cardinalities increase with object immigration.

Example 6.1 [17]**.** Let $(U,\mathscr{D}\cup U/d)$ be a covering decision information system, where $\mathscr{D}=\{\mathscr{C}_1,\mathscr{C}_2,\mathscr{C}_3,\mathscr{C}_4\}$, $\mathscr{C}_1=\{\{x_1,x_2,x_3,x_4\},\{x_5\}\}$, $\mathscr{C}_2=\{\{x_1,x_2\},\{x_3,x_4,x_5\}\}$, $\mathscr{C}_3=\{\{x_1,x_2,x_5\},\{x_3,x_4\}\}$, $\mathscr{C}_4=\{\{x_1,x_2\},\{x_3,x_4\},\{x_5\}\}$, and $U/d=\{D_1,D_2\}=\{\{x_1,x_2\},\{x_3,x_4,x_5\}\}$. By Definition 2.4, we obtain

$$\Gamma(\mathscr{D})=\begin{bmatrix}1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&1\end{bmatrix}.$$

By Definition 2.5, we have

$$\begin{aligned}\mathcal{X}_{SH(D_1)}&=\Gamma(\mathscr{D})\bullet\mathcal{X}_{D_1}=\begin{bmatrix}1&1&1&1&1\end{bmatrix}^T,\\ \mathcal{X}_{SL(D_1)}&=\Gamma(\mathscr{D})\odot\mathcal{X}_{D_1}=\begin{bmatrix}0&0&0&0&0\end{bmatrix}^T,\\ \mathcal{X}_{SH(D_2)}&=\Gamma(\mathscr{D})\bullet\mathcal{X}_{D_2}=\begin{bmatrix}1&1&1&1&1\end{bmatrix}^T,\\ \mathcal{X}_{SL(D_2)}&=\Gamma(\mathscr{D})\odot\mathcal{X}_{D_2}=\begin{bmatrix}0&0&0&0&0\end{bmatrix}^T.\end{aligned}$$

By Definition 2.4, we get

$$\Gamma(\mathscr{D}/\mathscr{C}_4)=\begin{bmatrix}1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&1\end{bmatrix},\quad \Gamma(\mathscr{D}/\{\mathscr{C}_2,\mathscr{C}_4\})=\begin{bmatrix}1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&0\\1&1&1&1&0\\1&1&0&0&1\end{bmatrix},$$

$$\Gamma(\mathscr{D}/\{\mathscr{C}_2,\mathscr{C}_3,\mathscr{C}_4\})=\begin{bmatrix}1&1&1&1&0\\1&1&1&1&0\\1&1&1&1&0\\1&1&1&1&0\\0&0&0&0&1\end{bmatrix},\quad \Gamma(\mathscr{D}/\{\mathscr{C}_1,\mathscr{C}_2,\mathscr{C}_4\})=\begin{bmatrix}1&1&0&0&1\\1&1&0&0&1\\0&0&1&1&0\\0&0&1&1&0\\1&1&0&0&1\end{bmatrix}.$$

By Definition 2.5, we derive

$$\begin{aligned}\mathcal{X}_{SH(D_1)}&=\Gamma(\mathscr{D}/\mathscr{C}_4)\bullet\mathcal{X}_{D_1},\mathcal{X}_{SL(D_1)}=\Gamma(\mathscr{D}/\mathscr{C}_4)\odot\mathcal{X}_{D_1},\\ \mathcal{X}_{SH(D_2)}&=\Gamma(\mathscr{D}/\mathscr{C}_4)\bullet\mathcal{X}_{D_2},\mathcal{X}_{SL(D_2)}=\Gamma(\mathscr{D}/\mathscr{C}_4)\odot\mathcal{X}_{D_2},\\ \mathcal{X}_{SH(D_1)}&=\Gamma(\mathscr{D}/\{\mathscr{C}_2,\mathscr{C}_4\})\bullet\mathcal{X}_{D_1},\mathcal{X}_{SL(D_1)}=\Gamma(\mathscr{D}/\{\mathscr{C}_2,\mathscr{C}_4\})\odot\mathcal{X}_{D_1},\\ \mathcal{X}_{SH(D_2)}&=\Gamma(\mathscr{D}/\{\mathscr{C}_2,\mathscr{C}_4\})\bullet\mathcal{X}_{D_2},\mathcal{X}_{SL(D_2)}=\Gamma(\mathscr{D}/\{\mathscr{C}_2,\mathscr{C}_4\})\odot\mathcal{X}_{D_2},\\ \mathcal{X}_{SH(D_1)}&\neq\Gamma(\mathscr{D}/\{\mathscr{C}_2,\mathscr{C}_3,\mathscr{C}_4\})\bullet\mathcal{X}_{D_1},\mathcal{X}_{SL(D_1)}=\Gamma(\mathscr{D}/\{\mathscr{C}_2,\mathscr{C}_3,\mathscr{C}_4\})\odot\mathcal{X}_{D_1},\\ \mathcal{X}_{SH(D_2)}&=\Gamma(\mathscr{D}/\{\mathscr{C}_2,\mathscr{C}_3,\mathscr{C}_4\})\bullet\mathcal{X}_{D_2},\mathcal{X}_{SL(D_2)}\neq\Gamma(\mathscr{D}/\{\mathscr{C}_2,\mathscr{C}_3,\mathscr{C}_4\})\odot\mathcal{X}_{D_2},\\ \mathcal{X}_{SH(D_1)}&\neq\Gamma(\mathscr{D}/\{\mathscr{C}_1,\mathscr{C}_2,\mathscr{C}_4\})\bullet\mathcal{X}_{D_1},\mathcal{X}_{SL(D_1)}=\Gamma(\mathscr{D}/\{\mathscr{C}_1,\mathscr{C}_2,\mathscr{C}_4\})\odot\mathcal{X}_{D_1},\\ \mathcal{X}_{SH(D_2)}&=\Gamma(\mathscr{D}/\{\mathscr{C}_1,\mathscr{C}_2,\mathscr{C}_4\})\bullet\mathcal{X}_{D_2},\mathcal{X}_{SL(D_2)}\neq\Gamma(\mathscr{D}/\{\mathscr{C}_1,\mathscr{C}_2,\mathscr{C}_4\})\odot\mathcal{X}_{D_2}.\end{aligned}$$

Therefore, we see that $\{\mathscr{C}_1,\mathscr{C}_3\}$ is a type-1 reduct of $(U,\mathscr{D}\cup U/d)$ by Definition 2.6.

Remark. In Example 6.1, there is a corresponding relation between the conditional attribute c_i in the information system $(U, C\cup\{d\})$, where $C=\{c_1,c_2,c_3,c_4\}$, and the covering $\mathscr{C}_i$ in the dynamic covering decision information system $(U,\mathscr{D}\cup U/d)$. Because $\{\mathscr{C}_1,\mathscr{C}_3\}$ is a type-1 reduct of $(U,\mathscr{D}\cup U/d)$, $\{c_1, c_3\}$ is an attribute reduct of the information system $(U, C\cup\{d\})$.

Example 6.2 (Continued from Example 6.1)**.** Let $(U^+,\mathscr{D}^+\cup U^+/d^+)$ be a dynamic covering decision information system of $(U,\mathscr{D}\cup U/d^+)$, where $\mathscr{D}^+=\{\mathscr{C}_1^+,\mathscr{C}_2^+,\mathscr{C}_3^+,\mathscr{C}_4^+\}$, $\mathscr{C}_1^+=\{\{x_1,x_2,x_3,x_4\},\{x_5\},\{x_6,x_7,x_8,x_9\}\}$, $\mathscr{C}_2^+=\{\{x_1,x_2\},\{x_3,x_4,x_5\},$

$\{x_6, x_7\}, \{x_8, x_9\}\}$, $\mathscr{C}_3^+ = \{\{x_1, x_2, x_5\}, \{x_3, x_4\}, \{x_6, x_7\}, \{x_8, x_9\}\}$, $\mathscr{C}_4^+ = \{\{x_1, x_2\}, \{x_3, x_4\}, \{x_5\}, \{x_6, x_7\}, \{x_8, x_9\}\}$, and $U^+/d^+ = \{D_1^+, D_2^+\} = \{\{x_1, x_2, x_6, x_7\}, \{x_3, x_4, x_5, x_8, x_9\}\}$. By Definition 2.4 and Theorem 3.3, we have

$$\Gamma(\mathscr{D}^+) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

By Definition 2.5, we get

$$\mathcal{X}_{SH(D_1^+)} = \Gamma(\mathscr{D}^+) \bullet \mathcal{X}_{D_1^+} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}^T,$$
$$\mathcal{X}_{SL(D_1^+)} = \Gamma(\mathscr{D}^+) \odot \mathcal{X}_{D_1^+} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T,$$
$$\mathcal{X}_{SH(D_2^+)} = \Gamma(\mathscr{D}^+) \bullet \mathcal{X}_{D_2^+} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}^T,$$
$$\mathcal{X}_{SL(D_2^+)} = \Gamma(\mathscr{D}^+) \odot \mathcal{X}_{D_2^+} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T.$$

By Theorem 3.3 and Example 6.1, we obtain

$$\Gamma(\mathscr{D}^+/\mathscr{C}_4^+) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix},$$

$$\Gamma(\mathscr{D}^+/\{\mathscr{C}_2^+, \mathscr{C}_4^+\}) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix},$$

$$\Gamma(\mathscr{D}^+/\{\mathscr{C}_2^+, \mathscr{C}_3^+, \mathscr{C}_4^+\}) = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix},$$

$$\Gamma(\mathscr{D}^+/\{\mathscr{C}_1^+, \mathscr{C}_2^+, \mathscr{C}_4^+\}) = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}.$$

By Definition 2.5, we have

$$\mathcal{X}_{SH(D_1^+)} = \Gamma(\mathscr{D}^+/\mathscr{C}_4^+) \bullet \mathcal{X}_{D_1^+}, \mathcal{X}_{SL(D_1^+)} = \Gamma(\mathscr{D}^+/\mathscr{C}_4^+) \odot \mathcal{X}_{D_1^+},$$

$$\begin{aligned}
\mathcal{X}_{SH(D_2^+)} &= \Gamma(\mathscr{D}^+/\mathscr{C}_4^+) \bullet \mathcal{X}_{D_2^+}, \mathcal{X}_{SL(D_2^+)} = \Gamma(\mathscr{D}^+/\mathscr{C}_4^+) \odot \mathcal{X}_{D_2^+},\\
\mathcal{X}_{SH(D_1^+)} &= \Gamma(\mathscr{D}^+/\{\mathscr{C}_2^+, \mathscr{C}_4^+\}) \bullet \mathcal{X}_{D_1^+}, \mathcal{X}_{SL(D_1^+)} = \Gamma(\mathscr{D}^+/\{\mathscr{C}_2^+, \mathscr{C}_4^+\}) \odot \mathcal{X}_{D_1^+},\\
\mathcal{X}_{SH(D_2^+)} &= \Gamma(\mathscr{D}^+/\{\mathscr{C}_2^+, \mathscr{C}_4^+\}) \bullet \mathcal{X}_{D_2^+}, \mathcal{X}_{SL(D_2^+)} = \Gamma(\mathscr{D}^+/\{\mathscr{C}_2^+, \mathscr{C}_4^+\}) \odot \mathcal{X}_{D_2^+},\\
\mathcal{X}_{SH(D_1^+)} &\neq \Gamma(\mathscr{D}^+/\{\mathscr{C}_2^+, \mathscr{C}_3^+, \mathscr{C}_4^+\}) \bullet \mathcal{X}_{D_1^+}, \mathcal{X}_{SL(D_1^+)} = \Gamma(\mathscr{D}^+/\{\mathscr{C}_2^+, \mathscr{C}_3^+, \mathscr{C}_4^+\}) \odot \mathcal{X}_{D_1^+},\\
\mathcal{X}_{SH(D_2^+)} &= \Gamma(\mathscr{D}^+/\{\mathscr{C}_2^+, \mathscr{C}_3^+, \mathscr{C}_4^+\}) \bullet \mathcal{X}_{D_2^+}, \mathcal{X}_{SL(D_2^+)} \neq \Gamma(\mathscr{D}^+/\{\mathscr{C}_2^+, \mathscr{C}_3^+, \mathscr{C}_4^+\}) \odot \mathcal{X}_{D_2^+},\\
\mathcal{X}_{SH(D_1^+)} &\neq \Gamma(\mathscr{D}^+/\{\mathscr{C}_1^+, \mathscr{C}_2^+, \mathscr{C}_4^+\}) \bullet \mathcal{X}_{D_1^+}, \mathcal{X}_{SL(D_1^+)} \neq \Gamma(\mathscr{D}^+/\{\mathscr{C}_1^+, \mathscr{C}_2^+, \mathscr{C}_4^+\}) \odot \mathcal{X}_{D_1^+},\\
\mathcal{X}_{SH(D_2^+)} &\neq \Gamma(\mathscr{D}^+/\{\mathscr{C}_1^+, \mathscr{C}_2^+, \mathscr{C}_4^+\}) \bullet \mathcal{X}_{D_2^+}, \mathcal{X}_{SL(D_2^+)} \neq \Gamma(\mathscr{D}^+/\{\mathscr{C}_1^+, \mathscr{C}_2^+, \mathscr{C}_4^+\}) \odot \mathcal{X}_{D_2^+}.
\end{aligned}$$

Therefore, we see that $\{\mathscr{C}_1^+, \mathscr{C}_3^+\}$ is a type-1 reduct of $(U^+, \mathscr{D}^+ \cup U^+/d^+)$ by Definition 2.6.

By Theorem 3.5, a type-2 reduct of $(U^+, \mathscr{D}^+ \cup U^+/d^+)$ can be constructed similarly as Examples 6.1 and 6.2. For simplicity, we do not present the process in this section.

Remark. In Example 6.2, there is a corresponding relation between the conditional attribute c_i^+ in the information system $(U^+, C \cup \{d\})$, where $C = \{c_1, c_2, c_3, c_4\}$, and the covering $\mathscr{C}_i^+$ in the dynamic covering decision information system $(U^+, \mathscr{D}^+ \cup U^+/d^+)$. Because $\{\mathscr{C}_1^+, \mathscr{C}_3^+\}$ is a type-1 reduct of $(U^+, \mathscr{D}^+ \cup U^+/d^+)$, $\{c_1, c_3\}$ is an attribute reduct of the information system $(U^+, C \cup \{d\})$.

7. Conclusions

In this paper, we provide efficient approaches to constructing the second and sixth lower and upper approximations of concepts in dynamic covering approximation spaces. Concretely, we use the incremental approach to constructing the type-1 and type-2 characteristic matrices of dynamic coverings whose cardinalities increase or decrease with varying object sets. We design the incremental algorithms to compute the second and sixth lower and upper approximations of sets. Experimental results illustrate that the computation of the second and sixth lower and upper set approximations is significantly faster when using the incremental approaches. Finally, we perform knowledge reduction of dynamic covering decision information systems while increasing the cardinalities of coverings.

In the future, we will propose more efficient approaches to constructing the type-1 and type-2 characteristic matrices of dynamic coverings. Additionally, we will focus on the development of efficient approaches for knowledge discovery in dynamic covering decision information systems.

Acknowledgments

We would like to thank the anonymous reviewers very much for their professional comments and valuable suggestions. This work is supported by the National Natural Science Foundation of China (nos.61273304, 61573255, 11201490, 11371130, 11401052, 11401195, 11526039, 11526038), Doctoral Fund of Ministry of Education of China (no. 201300721004), China Postdoctoral Science Foundation (nos. 2013M542558 and 2015M580353), the Scientific Research Fund of Hunan Provincial Education Department (nos. 14C0049 and 15B004).

References

[1] S. Asharaf, M. Murty, S. Shevade, Rough set based incremental clustering of interval data, Pattern Recognit. Lett. 27 (9) (2006) 515–519.
[2] X.X. Bian, P. Wang, Z.M. Yu, X.L. Bai, B. Chen, Characterizations of coverings for upper approximation operators being closure operators, Inf. Sci. 314 (2015) 41–54.
[3] H.M. Chen, T.R. Li, S.J. Qiao, D. Ruan, A rough set based dynamic maintenance approach for approximations in coarsening and refining attribute values, Int. J. Intell. Syst. 25 (10) (2010) 1005–1026.
[4] H.M. Chen, T.R. Li, D. Ruan, Maintenance of approximations in incomplete ordered decision systems while attribute values coarsening or refining, Knowl. Based Syst. 31 (2012) 140–161.
[5] H.M. Chen, T.R. Li, D. Ruan, J.H. Lin, C.X. Hu, A rough-set based incremental approach for updating approximations under dynamic maintenance environments, IEEE Trans. Knowl. Data Eng. 25 (2) (2013) 174–184.
[6] D.G. Chen, C.Z. Wang, Q.H. Hu, A new approach to attributes reduction of consistent and inconsistent covering decision systems with covering rough sets, Inf. Sci. 177 (17) (2007) 3500–3518.
[7] D.G. Chen, X.X. Zhang, W.L. Li, On measurements of covering rough sets based on granules and evidence theory, Inf. Sci. 317 (2015) 329–348.
[8] J.H. Dai, Rough set approach to incomplete numerical data, Inf. Sci. 241 (2013) 43–57.
[9] Y. Du, Q.H. Hu, P.F. Zhu, P.J. Ma, Rule learning for classification based on neighborhood covering reduction, Inf. Sci. 181 (24) (2011) 5457–5467.
[10] A. Frank, A. Asuncion, UCI Machine Learning Repository, University of California, School of Information and Computer Science, Irvine, CA, 2010. [http://archive.ics.uci.edu/ml].
[11] Y.N. Fan, T.L. Tseng, C.C. Chen, C.C. Huang, Rule induction based on an incremental rough set, Expert Syst. Appl. 36 (9) (2009) 11439–11450.
[12] L.H. Guan, G.Y. Wang, Generalized approximations defined by non-equivalence relations, Inf. Sci. 193 (15) (2012) 163–179.
[13] Q.H. Hu, D.R. Yu, Z.X. Xie, Neighborhood classifiers, Expert Syst. Appl. 34 (2) (2008) 866–876.
[14] C.C. Huang, T.L. Tseng, Y.N. Fan, C.H. Hsu, Alternative rule induction methods based on incremental object using rough set theory, Appl. Soft Comput. 13 (2013) 372–389.
[15] F. Jiang, Y.F. Sui, C.G. Cao, An incremental decision tree algorithm based on rough sets and its application in intrusion detection, Artif. Intell. Rev. 40 (2013) 517–530.
[16] P. Joshi, P. Kulkarni, Incremental learning: Areas and methods-a survey, Int. J. Data Min. Knowl. Manag. Process 2 (5) (2012) 43–51.

[17] G.M. Lang, Q.G. Li, M.J. Cai, T. Yang, Characteristic matrices-based knowledge reduction in dynamic covering decision information systems, Knowl. Based Syst. 85 (2015) 1–26.
[18] G.M. Lang, Q.G. Li, M.J. Cai, T. Yang, Q.M. Xiao, Incremental approaches to constructing approximations of sets based on characteristic matrices, Int. J. Mach. Learn. Cybern. (2014), doi:10.1007/s13042-014-0315-4.
[19] G.M. Lang, Q.G. Li, L.K. Guo, Homomorphisms between covering approximation spaces, Fundam. Inf. 137 (2015) 351–371.
[20] G.M. Lang, Q.G. Li, L.K. Guo, Homomorphisms-based attribute reduction of dynamic fuzzy covering information systems, Int. J. Gen. Syst. 44 (7-8) (2015) 791–811.
[21] G.M. Lang, Q.G. Li, T. Yang, An incremental approach to attribute reduction of dynamic set-valued information systems, Int. J. Mach. Learn. Cybern. 5 (2014) 775–788.
[22] Y. Leung, W.Z. Wu, W.X. Zhang, Knowledge acquisition in incomplete information systems: a rough set approach, Eur. J. Oper. Res. 168 (2006) 164–180.
[23] S.Y. Li, T.R. Li, D. Liu, Incremental updating approximations in dominance-based rough sets approach under the variation of the attribute set, Knowl. Based Syst. 40 (2013) 17–26.
[24] S.Y. Li, T.R. Li, D. Liu, Dynamic maintenance of approximations in dominance-based rough set approach under the variation of the object set, Int. J. Intell. Syst. 28 (8) (2013) 729–751.
[25] J.H. Li, C.L. Mei, Y.J. Lv, Incomplete decision contexts: Approximate concept construction, rule acquisition and knowledge reduction, Int. J. Approx. Reason. 54 (1) (2013) 149–165.
[26] T.R. Li, D. Ruan, W. Geert, J. Song, Y. Xu, A rough sets based characteristic relation approach for dynamic attribute generalization in data mining, Knowl. Based Syst. 20 (5) (2007) 485–494.
[27] T.R. Li, D. Ruan, J. Song, Dynamic maintenance of decision rules with rough set under characteristic relation, in: Proceedings of International Conference on Wireless Communications, Networking and Mobile Computing, Shanghai, China, 2007, pp. 3713–3716.
[28] Z.W. Li, N.X. Xie, G.Q. Wen, Soft coverings and their parameter reductions, Appl. Soft Comput. 31 (2015) 48–60.
[29] J.Y. Liang, F. Wang, C.Y. Dang, Y.H. Qian, A group incremental approach to feature selection applying rough set technique, IEEE Trans. Knowl. Data Eng. 26 (2) (2014) 294–308.
[30] G.L. Liu, The axiomatization of the rough set upper approximation operations, Fundam. Inf. 69 (3) (2006) 331–342.
[31] G.L. Liu, The relationship among different covering approximations, Inf. Sci. 250 (2013) 178–183.
[32] G.L. Liu, Special types of coverings and axiomatization of rough sets based on partial orders, Knowl. Based Syst. 85 (2015) 316–321.
[33] G.L. Liu, K. Zhu, The relationship among three types of rough approximation pairs, Knowl. Based Syst. 60 (2014) 28–34.
[34] D. Liu, T.R. Li, D. Ruan, J.B. Zhang, Incremental learning optimization on knowledge discovery in dynamic business intelligent systems, J. Glob. Optim. 51 (2) (2011) 325–344.
[35] D. Liu, T.R. Li, D. Ruan, W.L. Zou, An incremental approach for inducing knowledge from dynamic information systems, Fundam. Inf. 94 (2) (2009) 245–260.
[36] D. Liu, T.R. Li, J.B. Zhang, A rough set-based incremental approach for learning knowledge in dynamic incomplete information systems, Int. J. Approx. Reason. 55 (8) (2014) 1764–1786.
[37] D. Liu, T.R. Li, J.B. Zhang, Incremental updating approximations in probabilistic rough sets under the variation of attributes, Knowl. Based Syst. 73 (2015) 81–96.
[38] D. Liu, D.C. Liang, C.C. Wang, A novel three-way decision model based on incomplete information system, Knowl. Based Syst. 91 (2016) 32–45.
[39] C.H. Liu, D.Q. Miao, J. Qian, On multi-granulation covering rough sets, Int. J. Approx. Reason. 55 (6) (2014) 1404–1418.
[40] C. Luo, T.R. Li, H.M. Chen, Dynamic maintenance of approximations in set-valued ordered decision systems under the attribute generalization, Inf. Sci. 257 (2014) 210–228.
[41] C. Luo, T.R. Li, H.M. Chen, D. Liu, Incremental approaches for updating approximations in set-valued ordered information systems, Knowl. Based Syst. 50 (2013) 218–233.
[42] C. Luo, T.R. Li, H.M. Chen, L.X. Lu, Fast algorithms for computing rough approximations in set-valued decision systems while updating criteria values, Inf. Sci. 299 (2015) 221–242.
[43] D.Q. Miao, C. Gao, N. Zhang, Z.F. Zhang, Diverse reduct subspaces based co-training for partially labeled data, Int. J. Approx. Reason. 52 (8) (2011) 1103–1117.
[44] S. Ozawa, S. Pang, N. Kasabov, Incremental learning of chunk data for online pattern classification systems, IEEE Trans. Neural Netw. 19 (6) (2008) 1061–1074.
[45] Y.H. Qian, J.Y. Liang, D.Y. Li, F. Wang, N.N. Ma, Approximation reduction in inconsistent incomplete decision tables, Knowl. Based Syst. 23 (5) (2010) 427–433.
[46] M. Restrepo, C. Cornelis, J. Gmez, Partial order relation for approximation operators in covering based rough sets, Inf. Sci. 284 (2014) 44–59.
[47] P. Samanta, M.K. Chakraborty, Covering based approaches to rough sets and implications lattices, Lect. Notes Artif. Intell. 5908 (2009) 127–134.
[48] Y.L. Sang, J.Y. Liang, Y.H. Qian, Decision-theoretic rough sets under dynamic granulation, Knowl. Based Syst. 91 (2016) 84–92.
[49] M.W. Shao, Y. Leung, W.Z. Wu, Rule acquisition and complexity reduction in formal decision contexts, Int. J. Approx. Reason. 55 (2014) 259–274.
[50] N. Shan, W. Ziarko, Data-based acquisition and incremental modification of classification rules, Comput. Intell. 11 (2) (1995) 357–370.
[51] A. Skowron, C. Rauszer, The discernibility matrices and functions in information systems, in: R. Slowinski (Ed.), Intelligent Decision Support, Handbook of Applications and Advances of the Rough Sets Theory, Kluwer, Dordrecht, 1992.
[52] W.H. Shu, H. Shen, Updating attribute reduction in incomplete decision systems with the variation of attribute set, Int. J. Approx. Reason. 55 (3) (2013) 867–884.
[53] W.H. Shu, H. Shen, Incremental feature selection based on rough set in dynamic incomplete data, Pattern Recognit. 47 (12) (2014) 3890–3906.
[54] Y.R. Syau, E.B. Lin, Neighborhood systems and covering approximation spaces, Knowl. Based Syst. 66 (2014) 61–67.
[55] A.H. Tan, J.J. Li, Y.J. Lin, G.P. Lin, Matrix-based set approximations and reductions in covering decision information systems, Int. J. Approx. Reason. 59 (2015) 68–80.
[56] A.H. Tan, J.J. Li, G.P. Lin, Y.J. Lin, Fast approach to knowledge acquisition in covering information systems using matrix operations, Knowl. Based Syst. 79 (2015) 90–98.
[57] C.Z. Wang, Q. He, D.G. Chen, Q.H. Hu, A novel method for attribute reduction of covering decision systems, Inf. Sci. 254 (2014) 181–196.
[58] C.Z. Wang, M.W. Shao, B.Q. Sun, Q.H. Hu, An improved attribute reduction scheme with covering based rough sets, Appl. Soft Comput. 26 (2015) 235–243.
[59] F. Wang, J.Y. Liang, C.Y. Dang, Attribute reduction for dynamic data sets, Appl. Soft Comput. 13 (2013) 676–689.
[60] F. Wang, J.Y. Liang, Y.H. Qian, Attribute reduction: A dimension incremental strategy, Knowl. Based Syst. 39 (2013) 95–108.
[61] S.P. Wang, W. Zhu, Q.H. Zhu, F. Min, Characteristic matrix of covering and its application to boolean matrix decomposition and axiomatization, Inf. Sci. 263 (1) (2014) 186–197.
[62] Z.H. Wang, H. Wang, Q.R. Feng, L. Shu, The approximation number function and the characterization of covering approximation space, Inf. Sci. 305 (1) (2015) 196–207.
[63] W.Z. Wu, W.X. Zhang, Neighborhood operator systems and approximations, Inf. Sci. 144 (1) (2002) 201–217.
[64] W.H. Xu, W.X. Zhang, Measuring roughness of generalized rough sets induced by a covering, Fuzzy Sets Syst. 158 (22) (2007) 2443–2455.
[65] T. Yang, Q.G. Li, Reduction about approximation spaces of covering generalized rough sets, Int. J. Approx. Reason. 51 (3) (2010) 335–345.
[66] X.B. Yang, M. Zhang, H.L. Dou, J.Y. Yang, Neighborhood systems-based rough sets in incomplete information system, Knowl. Based Syst. 24 (6) (2011) 858–867.

[67] X.B. Yang, Y. Qi, H.L. Yu, X.N. Song, J.Y. Yang, Updating multigranulation rough approximations with increasing of granular structures, Knowl. Based Syst. 64 (2014) 59–69.
[68] Y.Y. Yao, Relational interpretations of neighborhood operators and rough set approximation operators, Inf. Sci. 111 (1) (1998) 239–259.
[69] Y.Y. Yao, B.X. Yao, Covering based rough set approximations, Inf. Sci. 200 (2012) 91–107.
[70] Y.Y. Yao, Y. Zhao, Discernibility matrix simplification for constructing attribute reducts, Inf. Sci. 179 (2009) 867–882.
[71] W. Zakowski, Approximations in the space (u, π), Demonstr. Math. 16 (1983) 761–769.
[72] J.B. Zhang, T.R. Li, H.M. Chen, Composite rough sets for dynamic data mining, Inf. Sci. 257 (2014) 81–100.
[73] J.B. Zhang, T.R. Li, D. Ruan, D. Liu, Rough sets based matric approaches with dynamic attribute variation in set-valued information systems, Int. J. Approx. Reason. 53 (4) (2012) 620–635.
[74] J.B. Zhang, T.R. Li, D. Ruan, D. Liu, Neighborhood rough sets for dynamic data mining, Int. J. Intell. Syst. 27 (4) (2012) 317–342.
[75] P. Zhu, Covering rough sets based on neighborhoods: an approach without using neighborhoods, Int. J. Approx. Reason. 52 (3) (2011) 461–472.
[76] W. Zhu, Relationship among basic concepts in covering-based rough sets, Inf. Sci. 179 (14) (2009) 2478–2486.
[77] W. Zhu, Relationship between generalized rough sets based on binary relation and coverings, Inf. Sci. 179 (3) (2009) 210–225.