

Dynamic Maximal Cliques Detection and Evolution Management in Social Internet of Things: A Formal Concept Analysis Approach

Yixuan Yang, Fei Hao, Beibei Pang, Geyong Min, and Yulei Wu

Abstract—The booming of Social Internet of Things (SIoT) has witnessed the significance of graph mining and analysis for social network management. Online Social Networks (OSNs) can be efficiently managed by monitoring users' behaviors within a cohesive social group represented by a maximal clique. They can further provide valued social intelligence for their users. Maximal Cliques Problem (MCP) as a fundamental problem in graph mining and analysis is to identify the maximal cliques in a graph. Existing studies on MCP mainly focus on static graphs, with less attention on the detection and dynamic evolution of maximal cliques in OSNs. To fill this gap, we adopt the Formal Concept Analysis (FCA) theory to represent and analyze social networks. We then develop two novel formal concepts generation algorithms, termed Add-FCA and Dec-FCA, that can be applicable to OSNs for detecting the maximal cliques and characterizing the dynamic evolution process of maximal cliques in OSNs. Extensive experimental results are conducted to investigate and demonstrate the correctness and effectiveness of the proposed algorithms. The results reveal that our algorithms can efficiently capture and manage the evolutionary patterns of maximal cliques, including unchanged, changed, added, and vanished maximal cliques in OSNs, and a quantitative relation among them is presented. In addition, an illustrative example is presented to verify the usefulness of the proposed approach.

Index Terms—Maximal Cliques Detection, Maximal Cliques Evolution, Online Social Networks, Formal Concept Analysis

I. INTRODUCTION

The Internet of Things (IoT) expands the communication between people to people, people to things, and things to things [1]. In recent years, with the emergence of smart objects, the potential for interaction between people and things, and between things and things has been further explored. The Social Internet of Things (SIoT) [2], combining Online Social Networks (OSNs) and IoT, has emerged as an important Internet application [3] and made the management of intelligent devices easier and safer for users [4], [5]. Many existing algorithms on OSNs, such as social network topological structure analysis, have been applied to the SIoT realm [6].

Y. Yang, F. Hao and B. Pang are with Key Laboratory of Modern Teaching Technology, Ministry of Education, Xi'an, China & School of Computer Science, Shaanxi Normal University, Xi'an, China. F. Hao is also with the Department of Computer Science, College of Engineering, Mathematics, and Physical Sciences, University of Exeter, Exeter, EX4 4QF, U.K.

G. Min and Y. Wu are with the Department of Computer Science, College of Engineering, Mathematics, and Physical Sciences, University of Exeter, Exeter, EX4 4QF, U.K.

*Corresponding author: Fei Hao; Email:feehao@gmail.com

Manuscript received****, ****; revised ****, ****.

With the advent of ubiquitous connectivity coupled with the proliferation of social network services and mobile/IoT devices, the number of users of OSNs has greatly increased in recent years. Mobile users have been using social network services such as Twitter, Facebook and Weibo to chat, obtain news and information, and share their videos, which enables convenient communication between users. Such mobile users essentially form social groups (cliques). Clique detection aims to explore the community structure of OSNs and monitor the behaviors of users within a community, so as to efficiently manage OSNs, such as security and privacy [7], content dissemination [8], mobile crowd sensing management [9], service composition [10], as well as rumors control [11]. The essential of these applications is to efficiently discover and manage the topological structures of OSNs. For example, there exist a large number of online water armies on Weibo, and they can potentially form a community. By managing the various changes in the structures of this community, we can predict whether an online water army is the potential rumors spreader, and further detect the target of their rumors, in order to prevent the spreading of rumors in time. The control of rumors is a kind of social network management. Wang *et al.* [11] attempted to control the spread of negative information by establishing a coupling diffusion model of positive and negative information, and derived the critical conditions for the negative information diffusion. Motivated by the above application, the aim of this paper is to detect the maximal cliques and obtain their evolutionary patterns in OSNs, in order to provide more valued OSNs services.

Maximal Cliques Problem (MCP), a classic combinatorial optimization problem in graph theory, is an NP-complete problem. In recent years, MCP has been widely used in market analysis, solution selection, signal transmission, computer vision, fault diagnosis, data mining and other fields. There exist a range of studies using maximal cliques to analyze and predict potential churn customers, apply data mining in grid systems [12], perform protein structure analysis and classification [13]–[15], predict the structure of proteins from molecular sequences [16], as well as investigate the functional relationships between proteins [17]. In particular, in [14], their method was based on the algorithm proposed by Bron and Kerbosch, which enumerates all maximal cliques in a graph. They restricted the search process to the cliques that represent connected substructures, and this can reduce the number of cliques to be considered during the search process and the size of the search tree drastically. In [16], maximal cliques

were used in a comparative modeling scenario to build side-chains, regions of main chain, and mix and match between different homologs in a context-sensitive manner.

MCP has also been applied to social network analysis. For instance, mining maximal cliques from social networks can obtain the groups with unique characteristics, where users can easily obtain targeted recommendation services or advertisement propagation. For example, in Weibo (an online social platform for information sharing, dissemination and acquisition based on user interaction), users select to become fans of other users according to their career, hobbies, hometown and other factors. We regard users as nodes in the social network graph, where the relationship between users is the edge between nodes. By analyzing the structure of the social network, we can detect several maximal cliques, among which some nodes are covered by several maximal cliques. These nodes are regarded as special key nodes and can be treated as the core characters in the social network with relatively extensive contacts. When these core characters upload or forward the blog in Weibo, the blog will be more easily seen by more users. If this blog is a rumor, when we need to refute the rumor, we can find these core characters easily by detecting the maximal cliques. The reason is that we can delete this Weibo for controlling the rumor spread, or to clarify the rumor. However, due to the activities of human interactions, the topology of social networks usually possesses the characteristics of time-varying changes. For example, the relationship between users will change; there may be new maximal cliques; some maximal cliques may disappear or change. All the these can lead to the change of core characters. Therefore, it is important to examine social networks dynamically.

There have been many approaches for detecting maximal cliques that can be applicable to social networks. For example, Kumar *et al.* [18] presented an algorithm for enumerating maximal cliques in large-scale social networks. Our recent work [6] proposed an approach to detect maximal clique from general static social networks using Formal Concept Analysis (FCA). There are also several studies on the detection of maximal cliques in OSNs. For example, Pan *et al.* [19] presented an anytime-anywhere approach for Maximal Cliques Enumeration (MCE). Different from the existing works, this paper adopts the FCA theory to address the problem of dynamic maximal cliques detection in OSNs. Specifically, two concept lattice generation algorithms, called *Add-FCA* and *Dec-FCA*, are developed for the cases that users join in and leave from OSNs, respectively.

Our main contributions are summarized as follows:

- **OSNs-oriented Concept Lattice Generation.** The *Add-FCA* and *Dec-FCA* algorithms are developed to generate concept lattice when the formal context is changed. The *Add-FCA* algorithm is devised when the objects and the attributes increase individually or simultaneously, to generate new concept lattice for the new updated formal context. The *Dec-FCA* algorithm is proposed to generate new concept lattice for the new updated formal context where the objects and the attributes decrease individually or simultaneously. Further, an OSNs-oriented concept

lattice generation algorithm which considers the dynamic formal context constructed by a given social network is presented.

- **Maximal Cliques Detection.** This paper skillfully switches from the analysis of graph theory to the study of special formal concepts in the concept lattice. Specifically, we present an equivalent theorem between equiconcepts and maximal cliques. Therefore, the problem of detecting maximal cliques after adding or deleting nodes from social networks can be transformed to the problem of identifying equiconcepts from formal concept lattices.
- **Maximal Cliques Evolution.** We utilize the public real datasets to carry out extensive experiments, and validate the correctness and feasibility of the proposed *Add-FCA* and *Dec-FCA* algorithms. Importantly, our algorithms can efficiently capture the evolutionary patterns of maximal cliques including unchanged, changed, added, and vanished maximal cliques in OSNs. With the observation patterns, a quantitative relation among them is then presented. In addition, aiming to reveal the usefulness of our algorithms, a real social network is used as a proof-of-concept study. Then, we use the proposed algorithms to calculate and analyze the maximal cliques evolution of this real social network.

The rest of this paper is organized as follows. Section II overviews the related work of concept lattice generation algorithms in FCA and FCA-based social network analysis. The preliminaries on the FCA theory, cliques and maximal cliques are provided in Section III. Section IV presents a detailed description of the proposed problem. Section V proposes the algorithms of *Add-FCA* and *Dec-FCA*, respectively. The detection approach of maximal cliques from OSNs and their evolution are elaborated in Section VI. Section VII presents the implementation details and carries out experimental results and analysis. Finally, Section VIII concludes this paper.

II. RELATED WORK

Since we plan to adopt FCA to study the detection and evolution of maximal cliques in OSNs, this section summarizes the existing research on algorithms of generating formal concept lattices and maximal cliques detection.

A. Concept Lattice Generation

FCA theory was proposed by Wille in 1982 [20]. It has played an important role in cognitive computing, pattern recognition, machine learning, decision-making analysis, web search, and community recommendation. However, the algorithm for generating concept lattices is a challenge and important research issue of FCA.

Most existing algorithms on concept lattice generation are applicable for general static formal contexts. For example, Ganter *et al.* [20] proposed a Next Closure algorithm for obtaining the concept lattice corresponding to a given formal context. The algorithm termed CMCG [21] was to generate the concept lattice by using the attribute rank of a concept matrix. The algorithms in both Nourine [22] and Chein [23] were also typical algorithms for general formal context. Ma

et al. [24] introduced the interval-set into the decision formal context and investigated the interval-set concept lattice and its attribute reduction method on deciding formal contexts. Qian *et al.* [25] studied the construction theory and interval-set concept lattice, and they further proposed an approach for constructing interval-set concept lattices on the basis of exploring the relationship between interval-set concept lattices and general concept lattices. He *et al.* [26] devised a method for generating attribute interval-set to describe incomplete information.

There are some concept lattice generation algorithms that deal with dynamic formal context. FastAddExtent [27] proposed a concept lattice generation algorithm when attributes are reduced, but the time complexity is high. Zhang *et al.* [28] put forward the concept lattice generation algorithm when attributes in the formal context are reduced, but this algorithm suffers a running time. The Fold [29] and Unfold [30] algorithms were proposed to zoom out to decrease or increase the granularity levels of attributes in FCA without rebuilding the new lattice, and they provided classification and preprocessing procedures that can improve algorithm efficiency. Yang *et al.* [31] proposed an algorithm for attribute-incremental formal context. However, the above algorithms mainly focus on the problem of deleting or adding attributes/objects, and some algorithms are not efficient in terms of time and space. There has not been any existing work on concept lattice generation that is suitable for a dynamic social network.

B. Maximal Clique Detection

Maximal cliques detection has been widely used in a set of fields, such as DNA detection and clique enumeration, and it can also tackle some critical problems of social networks. Das *et al.* [32] utilized parallel graph algorithms to enumerate maximal cliques for further investigation of biological networks. Cheng *et al.* [33] firstly proposed an efficient partition-based algorithm for MCE that addresses the problem of processing large graphs with limited memory. Bera *et al.* [34] presented an efficient algorithm for finding all maximal cliques from a trapezoid graph.

FCA methodology has been used for topological structures mining from social networks. For example, in our previous work [6], we proposed a method to detect maximal cliques based on FCA. Snasel *et al.* [35] proposed the idea of using objects in social networks as a formal context, and the relationship between nodes as an inter-object relationship. Afterwards, they adopted FCA to address this problem of complex things in social networks [36]. Krajci *et al.* [37] also presented the idea of using FCA to deal with social networks. Unfortunately, the implementation details of their study were not provided. Hao *et al.* [38], [39] constructed the formal context by taking the nodes in social networks as the objects and attributes of the formal context; importantly, the k -equiconcept in concept lattice is defined, and the equivalent theorem between the k -equiconcept and k -clique is presented for detecting k -cliques efficiently.

In summary, the research on FCA-based OSNs analysis is still in its infancy. As the form of social networks has

undergone tremendous changes, tapping into the special structure of social networks has become a hot research area. We can obtain useful information from the special structure we have discovered. Social networks are essentially a dynamically changing network. It is valuable and important to investigate the dynamic maximal cliques detection with FCA in OSNs.

III. PRELIMINARIES

A. Formal Concept Analysis

FCA is a typical data analysis method. It defines a set of formal concepts that represent the relationships between objects and attributes in an information system. These formal concepts are organized as concept lattices with a partial order. Unlike traditional methods of data analysis, FCA focuses on human cognition and understanding of information systems. It has been widely used in data mining, machine learning, software engineering, data analysis, ontology and other fields.

In this paper, we intend to use FCA to analyze the structure of social networks according to the method proposed in [38], based on which we present a set of definitions in this section.

Definition 1: (Formal Context) A formal context is organized as a triple $K = (A, B, R)$, where $A = \{x_1, x_2, \dots, x_n\}$ is the set of objects, $B = \{y_1, y_2, \dots, y_m\}$ is the set of attributes, and R is the binary relation between A and B , where $R \subseteq A \times B$. $(x, y) \in R$ denotes that object x has the attribute y , and $(x, y) \notin R$ denotes that object x does not have the attribute y , where $x \in A, y \in B$. Let “1” denote $(x, y) \in R$ and “0” denote $(x, y) \notin R$. Then, this formal context can be viewed as an information system with only “0” or “1”.

$$\begin{cases} 1 & (x_i, y_j) \in R \\ 0 & (x_i, y_j) \notin R \end{cases}$$

Definition 2: (Operators \uparrow and \downarrow) Given a formal context $K = (A, B, R)$, \uparrow and \downarrow on $X \subseteq A$ and $Y \subseteq B$ are respectively defined as

$$X^\uparrow = \{y \in B | \forall x \in A, (x, y) \in R\} \quad (1)$$

$$Y^\downarrow = \{x \in A | \forall y \in B, (x, y) \in R\} \quad (2)$$

For $\forall x \in A$, let $\{x\}^\uparrow = x^\uparrow$. For $\forall y \in B$, let $\{y\}^\downarrow = y^\downarrow$.

Definition 3: (Concept) Given a formal context $K = (A, B, R)$, if (X, Y) satisfies $X^\uparrow = Y$ and $Y^\downarrow = X$, (X, Y) is called a concept, and X is the extent of the concept and Y is the intent of the concept. $\theta(K)$ is an operation for obtaining a concept from formal context K .

Definition 4: (Concept Lattice) For a set of all concepts of the formal context $K = (A, B, R)$, denoted as a Concept Lattice, $\Omega(K)$, $\theta(K) \subseteq \Omega(K)$ and $\Omega(K) = \sum \theta(K)$. $\phi(K)$ is a set of extent for all concepts in $\Omega(K)$, and $\psi(K)$ is a set of intent for all concepts in $\Omega(K)$.

In what follows, we provide an example to facilitate the understanding of a formal context.

Example 1: Table I shows a formal context K . The set of objects is $A = (x_1, x_2, x_3, x_4)$, and the set of attributes is $B = (y_1, y_2, y_3, y_4, y_5)$. “1” denotes that there exists the binary relation between A and B . For example, the object “ x_2 ” has the attributes “ y_1 ”, “ y_2 ”, and “ y_3 ”.

Fig. 1 illustrates the concept lattice for the context of Table I. As shown in Fig. 1, each node indicates a formal concept, and the labels of a node represent intents and extents of the concepts.

TABLE I
A FORMAL CONTEXT K .

$A \times B$	y_1	y_2	y_3	y_4	y_5
x_1	1	1	0	1	1
x_2	1	1	1	0	0
x_3	0	0	0	0	1
x_4	1	1	1	0	0

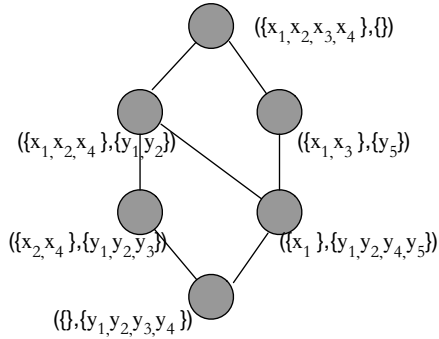


Fig. 1. The Concept Lattice of a Formal Context K .

B. Clique and Maximal Clique

Definition 5: (Clique) [38] Let $G = (V, E)$ be an undirected graph. A clique in G is a subset $S \subseteq V$ such that for any two vertices $v_i, v_j \in S$, there exists an edge $(v_i, v_j) \in E$.

Definition 6: (k -Clique) [38] Let $G = (V, E)$ be an undirected graph. A k -clique in G is a subset $S \subseteq V$ and $|S| = k$ such that for any two vertices $v_i, v_j \in S$, there exists an edge $(v_i, v_j) \in E$.

Definition 7: (Maximal Clique) [38] Maximal clique is a special clique which cannot be extended by including one more adjacent vertex. In other words, it is a clique which does not exist exclusively within the vertex set of a large clique. A subgraph H of a graph G is a maximal clique in G , if H is isomorphic to complete graph, and there is no vertex $v \in V(G)/V(H)$ so that v is adjacent to each vertex of H .

IV. PROBLEM DEFINITION

No matter in the virtual network world or in the real-world networks, people will have certain social activities and form certain social relations among people, as shown in Fig. 2. Let us represent the social relations between users as a social network, in which each node denotes a user and each edge indicates the social relationship between users. As time elapses, some current users may leave the social network, and some new users may join the social network, resulting in the time-varying changes of the structure, and also the maximal clique, of a social network.

With the existing FCA-based maximal cliques detection approaches that have been developed for a static social

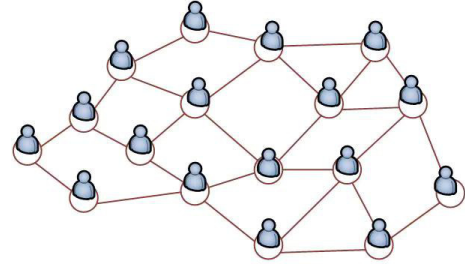


Fig. 2. A Social Network.

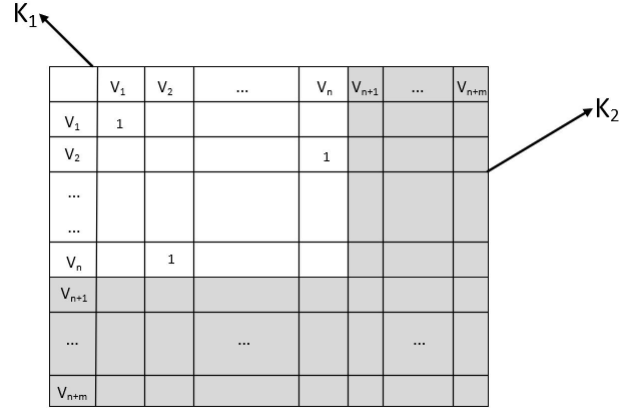


Fig. 3. User-incremental Formal Context for Adding Users.

network, we can rebuild a new formal context to update and extract maximal cliques. However, rebuilding the formal context requires high computational and time complexity, due to the large-scale and dynamical properties of OSNs. To solve this problem, in this paper we devise concept lattice generation algorithms for the formal context constructed from OSNs and analyze the relation between formal concepts and the structure of OSNs. Addressing this challenging problem requires the solutions of the following three sub-problems:

Sub-problem 1: Devising a concept lattice generation algorithm for the user-incremental formal context constructed from OSNs, where only users are added/joined. According to our previous work [38], we know that the objects (users) and attributes (of users) will be simultaneously added to the original formal context K_1 and updated as $K_1 \cup K_2$, as shown in Fig. 3. To generate the concept lattice from OSNs, the attribute-incremental (Fig. 4(a)) and object-incremental (Fig. 4(b)) concept lattice generation algorithms should be investigated first.

Sub-problem 2: Devising a concept lattice generation algorithm for the user-decremental formal context constructed from OSNs, where only users are deleted/left. Similar to Sub-problem 1, the objects (users) and attributes (of users) are simultaneously deleted from the original formal context K_1 and updated as $K_1 - K_2$, as shown in Fig. 5. To generate the concept lattice from OSNs, the attribute-decremental (Fig. 6(a)) and object-decremental (Fig. 6(b)) concept lattice generation algorithms should be investigated first.

	A_1	A_2	...	A_n	A_{n+1}	...	A_{n+m}
O_1	1						
O_2					1		
...							
...							
O_n		1					

(a) Attribute-incremental Formal Context

	A_1	A_2	...	A_n	A_{n+1}	...	A_{n+m}
O_1	1						
O_2					1		
...							
...							
O_n		1					

(a) Attribute-decremental Formal Context

	A_1	A_2	...	A_n
O_1	1			
O_2				1
...				
...				
O_n		1		
O_{n+1}				
...				
O_{n+m}				

(b) Object-incremental Formal Context

	A_1	A_2	...	A_n
O_1	1			
O_2				1
...				
...				
O_n		1		
O_{n+1}				
...				
O_{n+m}				

(b) Object-decremental Formal Context

Fig. 4. Attribute/Object-incremental Formal Contexts for Adding Users.

Fig. 6. Attribute/Object-decremental Formal Contexts for Removing Users.

	V_1	V_2	...	V_n	V_{n+1}	...	V_{n+m}
V_1	1						
V_2					1		
...							
...							
V_n		1					
V_{n+1}							
...							
V_{n+m}							

Fig. 5. User-decremental Formal Context for Removing Users.

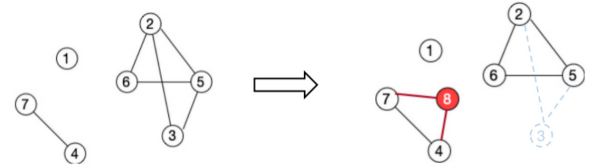


Fig. 7. Maximal Cliques Evolution in OSNs.

Sub-problem 3: Analyzing the relation between the formal concepts and the structure of OSNs for detecting the maximal cliques and capturing the evolution patterns (as shown in Fig. 7) by using the above concept lattice generation algorithms proposed in Sub-problems 1 and 2.

V. THE PROPOSED ADD-FCA AND DEC-FCA ALGORITHMS

To address the problems mentioned in the previous section, we present two types of concept lattice generation algorithms including object/attribute-incremental algorithms and object/attribute-decremental algorithms (Hereinafter referred to as Add-FCA and Dec-FCA, respectively).

A. The Add-FCA Algorithm

Reference [31] is our preliminary work. It has implemented the Algorithm for generating Attribute-Incremental Concept Lattice. The Add-FCA Algorithm in this paper is different from reference [31], which includes the Attribute-Incremental Concept Lattice Algorithm (**Algorithm 1**) and the Object-incremental Concept Lattice Generation Algorithm (**Algorithm 3**).

1) *Attribute-Incremental Concept Lattice Generation Algorithm:* In our previous work [31], a fast concept lattice generation algorithm for attribute-incremental formal context has been proposed and its correctness has been formally proved.

Let us consider a problem that a formal context $K_1=(A, B_1, R_1)$ is dynamically updated by adding a new formal context $K_2=(A, B_2, R_2)$, as shown in Fig. 4(a), and finally reach the formal context $K=(A, B, R)$, where $A=A$, $B= B_1 \cup B_2$, and $R=R_1 \cup R_2$. The main idea of the solution is that: our algorithm is to preserve the previously obtained concept lattice $\Omega(K_1)$ and generate the concept lattice $\Omega(K_2)$ of the new formal context K_2 . Then, we make the intersection

of $\phi(K_1)$ and $\phi(K_2)$, i.e., $\phi(K_1) \cap \phi(K_2)$, and store it into a set $\phi(K)$. After that, we obtain the intent i for each extent $e \in \phi(K)$ via $i \leftarrow e^\uparrow$. This working process is depicted in Algorithm 1.

Algorithm 1 Attribute-Incremental Concept Lattice Generation Algorithm

Input:

$$K_1 = (A, B_1, R_1)$$

$$K_2 = (A, B_2, R_2)$$

Output:

Set of concepts $\Omega(K)$

- 1: Initialize $\Omega(K_1) = \emptyset, \Omega(K_2) = \emptyset, \Omega(K) = \emptyset, \phi(K_1) = \emptyset, \phi(K_2) = \emptyset, \phi(K) = \emptyset$
 - 2: **begin**
 - 3: $\Omega(K_1) \leftarrow \text{BasicConcept}(K_1)$
 - 4: $\Omega(K_2) \leftarrow \text{BasicConcept}(K_2)$
 - 5: **for** each $\theta(X, Y) \in \Omega(K_1)$
 - 6: $\phi(K_1) \leftarrow X \cup \phi(K_1)$
 - 7: **end**
 - 8: **for** each $\theta(X, Y) \in \Omega(K_2)$
 - 9: $\phi(K_2) \leftarrow X \cup \phi(K_2)$
 - 10: **end**
 - 11: $\phi(K) \leftarrow \phi(K_1) \cap \phi(K_2)$
 - 12: **for** each extent $e \in \phi(K)$
 - 13: $i \leftarrow e^\uparrow$
 - 14: $\Omega(K) \leftarrow (e, i) \cup \Omega(K)$
 - 15: **return** $\Omega(K)$
 - 16: **end**
-

Line 1 initializes the sets of formal concepts $\Omega(K_1), \Omega(K_2), \Omega(K), \phi(K_1), \phi(K_2), \phi(K)$. Lines 2-5 respectively generate concept lattices of the formal context K_1 and K_2 by revoking the algorithm **BasicConcept** (as shown in Algorithm 2) which is a baseline algorithm for generating concept lattice. Lines 6-8 store all extents of $\theta(K_1)$ into the set $\Omega(K_1)$. Lines 9-11 store all extents of $\theta(K_2)$ into the set $\Omega(K_2)$. Line 12 takes the intersection of $\theta(K_1)$ and $\theta(K_2)$, and stores them in the set $\theta(K)$. Lines 13-16 traverse $\theta(K)$ and find the corresponding intent of each extent, and store it in the set $\Omega(K)$. Finally, the set $\Omega(K)$ returns.

Algorithm 2 BasicConcept

Input:

$$K = (A, B, R)$$

Output:

Set of concepts $\Omega(K)$

- 1: Initialize $\theta(K) = \emptyset, \Omega(K) = \emptyset$
 - 2: **begin**
 - 3: **for** each attribute $y \in B$
 - 4: $\theta(K) \leftarrow y^\downarrow \cup \theta(K)$
 - 5: **end**
 - 6: **for** each extent $e \in \theta(K)$
 - 7: $i \leftarrow e^\uparrow$
 - 8: $\Omega(K) \leftarrow (e, i) \cup \Omega(K)$
 - 9: **return** $\Omega(K)$
 - 10: **end**
-

2) *Object-Incremental Concept Lattice Generation Algorithm*: Similar to algorithm 1, we also devise an object-incremental concept lattice generation algorithm.

Let us consider a problem that a formal context $K_1=(A_1, B, R_1)$ is dynamically updated by adding a new formal context $K_2=(A_2, B, R_2)$, as shown in Fig. 4(b), and finally reach the formal context $K=(A, B, R)$, where $A=A_1 \cup A_2, B=B$, and $R=R_1 \cup R_2$. The main idea of the solution is that: our algorithm is to preserve the previously obtained concept lattice $\Omega(K_1)$ and generate the concept lattice $\Omega(K_2)$ of the new formal context K_2 . Then, we make the intersection of $\psi(K_1)$ and $\psi(K_2)$, i.e., $\psi(K_1) \cap \psi(K_2)$, and store it into a set $\psi(K)$. After that, we obtain the extent e for each intent $i \in \psi(K)$ via $e \leftarrow i^\downarrow$.

Theorem 1: For three given formal contexts $K_1 = \{A_1, B, R_1\}, K_2 = \{A_2, B, R_2\}, K = (A_1 \cup A_2, B, R_1 \cup R_2)$, the relation among the set of the intents $\psi(K_1), \psi(K_2)$, and $\psi(K)$ satisfies the following equation:

$$\psi(K) = \{Y_1 \cap Y_2; Y_1 \in \psi(K_1), Y_2 \in \psi(K_2)\} \quad (3)$$

Proof

- 1) Y_1 is an intent of concepts of $K_1, Y_1 \in \psi(K_1), Y_2$ is an intent of concepts of $K_2, Y_2 \in \psi(K_2), \exists X_1 \subseteq A_1$, then (X_1, Y_1) is a concept and $(X_1, Y_1) \in \Omega(K_1), \exists X_2 \subseteq A_2$, then (X_2, Y_2) is a concept, $(X_2, Y_2) \in \Omega(K_2)$. $Y_1 \cap Y_2 = X_1^\uparrow \cap X_2^\uparrow = (X_1 \cup X_2)^\uparrow$, due to $X_1 \cup X_2 \subseteq A_1 \cup A_2$, we have $((X_1 \cap X_2)^\uparrow, (Y_1 \cap Y_2)) = ((X_1 \cap X_2)^\uparrow, (X_1 \cap X_2)^\uparrow) = \Omega(K)$, hence, $Y_1 \cap Y_2 \subseteq \psi(K)$, the intersection of Y_1 and Y_2 is the intent of concepts of K . Besides, $Y \in \psi(K), \exists Y \subseteq B_1 \cup B_2$, then $(X, Y) \in \Omega(K), Y = X^\uparrow = (X \cap (A_1 \cup A_2))^\uparrow = ((X \cap A_1) \cup (X \cap A_2))^\uparrow = (X \cap A_1)^\uparrow \cap (X \cap A_2)^\uparrow$, due to $X \cap A_1 \subseteq A_1$, we have $(X \cap A_1)^\uparrow \in \psi(K_1)$, and $X \cap A_2 \subseteq A_2$ we have $(X \cap A_2)^\uparrow \in \psi(K_2)$, therefore, $\psi(K) = \{Y_1 \cap Y_2; Y_1 \in \psi(K_1), Y_2 \in \psi(K_2)\}$.
- 2) When $X_2 = \{n\}, K_2 = \{\{n\}, B, R_2\}, \psi(A \cup \{n\}, B, R) = \psi(A, B, R) \cup \{Y \cap n^\uparrow, Y \in \psi(A, B, R)\}$, according to 1), $\psi(\{n\}, B, R_2) = \{n\}^\uparrow$.

□

The above working process is depicted in Algorithm 3. In this algorithm, Line 1 initializes the sets $\Omega(K_1), \Omega(K_2), \Omega(K), \psi(K_1), \psi(K_2), \psi(K)$. Lines 2-5 respectively generate the concept lattices of formal context $\Omega(K_1)$ and $\Omega(K_2)$ via the Algorithm 2 **BasicConcept**. Lines 6-8 store all intents of $\Omega(K_1)$ into the set $\psi(K_1)$. Lines 9-11 store all intents of $\Omega(K_2)$ into the set $\psi(K_2)$. Line 12 takes the intersection of $\psi(K_1)$ and $\psi(K_2)$ and stores them in the set $\psi(K)$. Lines 13-16 traverse $\psi(K)$ and find the corresponding extent of each intent, and store them in the set $\Omega(K)$. Finally, the set $\Omega(K)$ returns.

For Algorithm 4, Line 1 initializes a set of concepts $\Omega(K_2)'$ and a set of concepts $\Omega(K)$. Line 3 obtains a new formal context $(K_2)'$ which only adds attributes. Line 4 generates a concept lattice $\Omega(K_2)'$ by Algorithm 1 and preserves $\Omega(K_2)'$. Line 5 obtains a new formal context $(K_2)''$ which adds objects in the formal context $(K_2)'$. Line 6 generates a concept lattice $\Omega(K)$ by Algorithm 3. Line 8 finally returns the set $\Omega(K)$.

Algorithm 3 Object-Incremental Concept Lattice Generation Algorithm

Input:

$$K_1 = (A_1, B, R_1)$$

$$K_2 = (A_1, B, R_2)$$

Output:Set of concepts $\Omega(K)$

```

1: Initialize  $\Omega(K_1) = \emptyset, \Omega(K_2) = \emptyset, \Omega(K) = \emptyset, \psi(K_1) = \emptyset, \psi(K_2) = \emptyset, \psi(K) = \emptyset$ 
2: begin
3:    $\Omega(K_1) \leftarrow \text{BasicConcept}(K_1)$ 
4:    $\Omega(K_2) \leftarrow \text{BasicConcept}(K_2)$ 
5: end
6: for each  $\theta(X, Y) \in \Omega(K_1)$ 
7:    $\psi(K_1) \leftarrow X \cup \psi(K_1)$ 
8: end
9: for each  $\theta(X, Y) \in \Omega(K_2)$ 
10:   $\psi(K_2) \leftarrow X \cup \psi(K_2)$ 
11: end
12:  $\psi(K) \leftarrow \psi(K_1) \cap \psi(K_2)$ 
13: for each intent  $i \in \psi(K)$ 
14:    $e \leftarrow i^\downarrow$ 
15:   if  $e \uparrow = i$ 
16:      $\Omega(K) \leftarrow (e, i) \cup \Omega(K)$ 
17:   else:
18:     break;
19:   return  $\Omega(K)$ 
20: end

```

Algorithm 4 Add-FCA

Input:

$$K_1 = (A_1, B_1, R_1)$$

$$K_2 = (A_2, B_2, R_2)$$

Output:Set of concepts $\Omega(K)$

```

1: Initialize  $\Omega(K_2)' = \emptyset, \Omega(K) = \emptyset$ 
2: begin
3:    $K_2' \leftarrow (A_1, B_1 \cup B_2, R_1 \cup R_2)$ 
4:    $\Omega(K_2)' \leftarrow \text{Attribute-Incremental Concept Lattice Generation Algorithm}(K_1, K_2')$ 
5:    $K_2'' \leftarrow (A_1 \cup A_2, B_1 \cup B_2, R_1 \cup R_2)$ 
6:    $\Omega(K) \leftarrow \text{Object-Incremental Concept Lattice Generation Algorithm}(K_2', K_2'')$ 
7:   return  $\Omega(K)$ 
8: end

```

The time complexities of Algorithm 1, Algorithm 2 and Algorithm 3 are $O(n^2 + n^3)$, $O(n + n^3)$ and $O(n^2 + n^3)$, respectively. Hence, the time complexity of Algorithm 4 is $O(n^2 + n^3) + O(n^2 + n^3) + O(n + n^2)$, which is $O(n^3)$.

For the **BasicConcept** algorithm, if the attributes or objects are added, the formal context needs to be updated and the corresponding concept lattice also needs to be built. Therefore, the time for generating the concept lattice is twice of Algorithm 4. Particularly, it consumes much more time to generate the new concept lattice for a formal context including a large number of objects or attributes.

Obviously, when users are added in OSNs, the objects (users) and attributes (of users) in the formal context will increase at the same time. When users are added in OSNs, we adopt Algorithm 4 to generate the corresponding concept lattice.

B. The Dec-FCA Algorithm

When users leave from OSNs, both objects (users) and attributes (users) in the formal context are also decreased accordingly.

Let us consider a problem that a formal context $K_1=(A_1, B_1, R_1)$ is dynamically updated by deleting the partial formal context $K_2=(A_2, B_2, R_2)$, as shown in Fig. 5, and finally reach the formal context $K=(A, B, R)$, where $A=A_1-A_2$, $B=B_1-B_2$, $R=R_1-R_2$. The solution idea is described as follows. Our algorithm is to update the objects set/attributes set as A'/B' by deleting all objects /attributes in K_2 from K_1 , respectively. Then, we can obtain a new formal context K and generate the concept lattice by revoking the Algorithm 2 **BasicConcept**.

This working process is depicted in Algorithm 5. In this algorithm, Line 1 initializes a set of formal concepts $\Omega(K)$, a set of object A , and a set of Attribute B . Lines 3-6 are in charge of deleting the formal context K_2 from the formal context K_1 . Then, the resulting formal context K is obtained. Line 7 generates the concept lattice by revoking the Algorithm 2 **BasicConcept**. Line 8 finally returns the set $\Omega(K)$.

Algorithm 5 Dec-FCA

Input:

$$K_1 = (A_1, B_1, R_1)$$

$$K_2 = (A_2, B_2, R_2)$$

Output:Set of concepts $\Omega(K)$

```

1: Initialize  $\Omega(K) = \emptyset$ , a set of objects  $A$ , a set of objects  $B$ , a set of objects  $R$ 
2: begin
3:    $A \leftarrow A_1 - A_2$ 
4:    $B \leftarrow B_1 - B_2$ 
5:    $R \leftarrow R_1 - R_2$ 
6:    $K \leftarrow (A, B, R)$ 
7:    $\Omega(K) \leftarrow \text{BasicConcept}(K)$ 
8:   return  $\Omega(K)$ 
9: end

```

 VI. MAXIMAL CLIQUES DETECTION AND EVOLUTION

A. Formal Context Construction

A social network g is formulated as a graph with the vertices indicating a set of individuals and the edges representing the relations between vertices. In this paper, we adopt the modified adjacency matrix to represent the formal context of g , that is, $FC(g)=(V, V, I)$, where I is the binary relation between two vertices.

Definition 8: (Modified Adjacency Matrix) [38] Let a social network be a graph with n vertices that are assumed

to be ordered from v_1 to v_n . The $n \times n$ matrix M is called a modified adjacency matrix, in which

$$M = \begin{cases} m_{ij} = 1 & \text{if } (v_i, v_j) \in E \\ m_{ij} = 1 & \text{if } i = j \\ m_{ij} = 0 & \text{otherwise} \end{cases}$$

We can easily construct the following formal context (Table II) of a social network g as shown in Fig. 7 by using Definition 8. Further, the corresponding concept lattice as shown in Fig. 8, is built by using the Algorithm 2 **BasicConcept**.

TABLE II
THE FORMAL CONTEXT OF A SOCIAL NETWORK g .

$V \times V$	V_1	V_2	V_3	V_4	V_5	V_6	V_7
V_1	1	0	0	0	0	0	0
V_2	0	1	1	0	1	1	0
V_3	0	1	1	0	1	0	0
V_4	0	0	0	1	0	0	1
V_5	0	1	1	0	1	1	0
V_6	0	1	0	0	1	1	0
V_7	0	0	0	1	0	0	1

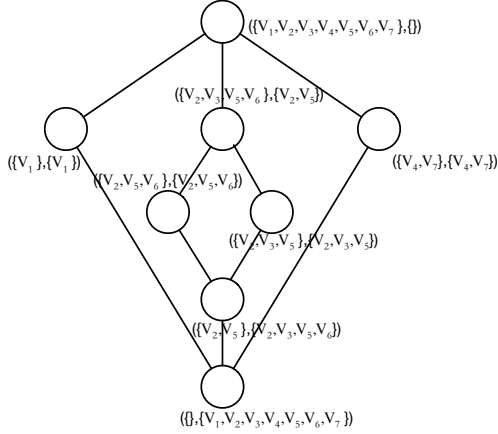


Fig. 8. Concept Lattice of a Social Network g .

For OSNs, in the event of users adding/leaving, a concept lattice of user-increment can be obtained by Algorithm 4 and a concept lattice of users-decrement can be obtained by Algorithm 5. There are some users added and some users removed at the same time, we can use Algorithm 4 and Algorithm 5, respectively, to obtain the concept lattice of OSNs.

B. Maximal Cliques Detection

In our previous work [6], we have proposed and proved the equivalence between the equiconcept and maximal clique.

Definition 9: (Equiconcept) [38] For a formal context $K=(A, B, R)$, if a pair (X, Y) satisfies $X^\uparrow = Y, Y^\downarrow = X$ and $X = Y$, the pair (X, Y) is an equiconcept, where X is called the extent of the equiconcept, and Y is called the intent of the equiconcept. Let $\beta(K)$ be the set of all equiconcepts with respect to the formal context K . As shown in Fig. 9, the nodes circled in red are all equiconcepts.

Theorem 2: [6] Given a social network G , the maximal cliques detection problem is equivalent to finding $\beta(FC(G))$.

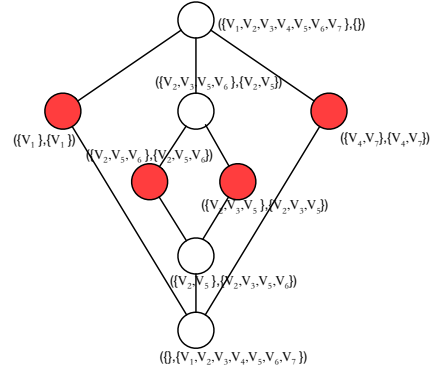


Fig. 9. The Equiconcepts of a Concept Lattice.

Proof The proof of the above theorem can be found in [6]. \square

Example 2: The visualization of $\Omega(\theta(FC(g)), \leq)$ is shown in Fig. 10, and the cliques circled are all maximal cliques. We can easily identify the following 4 maximal cliques, i.e., $\{1\}$, $\{4, 7\}$, $\{2, 3, 5\}$, $\{2, 5, 6\}$.

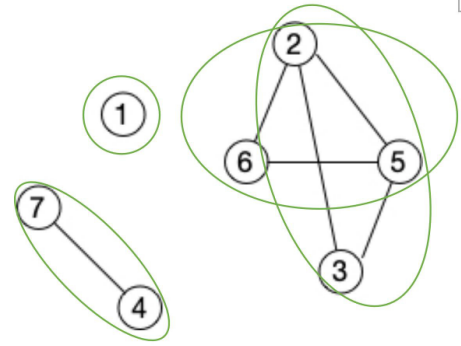


Fig. 10. The Maximal Cliques Detection of a Social Network g .

C. Maximal Cliques Evolution

This section mainly explores the evolution of maximal cliques in OSNs.

Example 3: In a social network g , as shown in Fig. 7, there are originally 7 nodes, and the corresponding concept lattice c is built as shown in Fig. 11. When the blue node 3 of the social network g is deleted, and the red nodes node 8 and node 9 are added, the corresponding concept lattice is represented as c' in Fig. 11.

Comparing equiconcepts in c and c' , the following important observations are obtained:

- **Observation 1:** When node 3 is deleted from g , $(\{2, 3, 5\}, \{2, 3, 5\})$ disappears from c . This procedure is described with a blue node and a blue dotted line.
- **Observation 2:** When node 8 is added in g , $(\{4, 7\}, \{4, 7\})$ (purple node) in c is changed to $(\{4, 7, 8\}, \{4, 7, 8\})$ in c' . This change is visualized with purple solid line.
- **Observation 3:** When node 9 is added in g , $(\{2, 6, 9\}, \{2, 6, 9\})$ is generated in c' but not in c . This

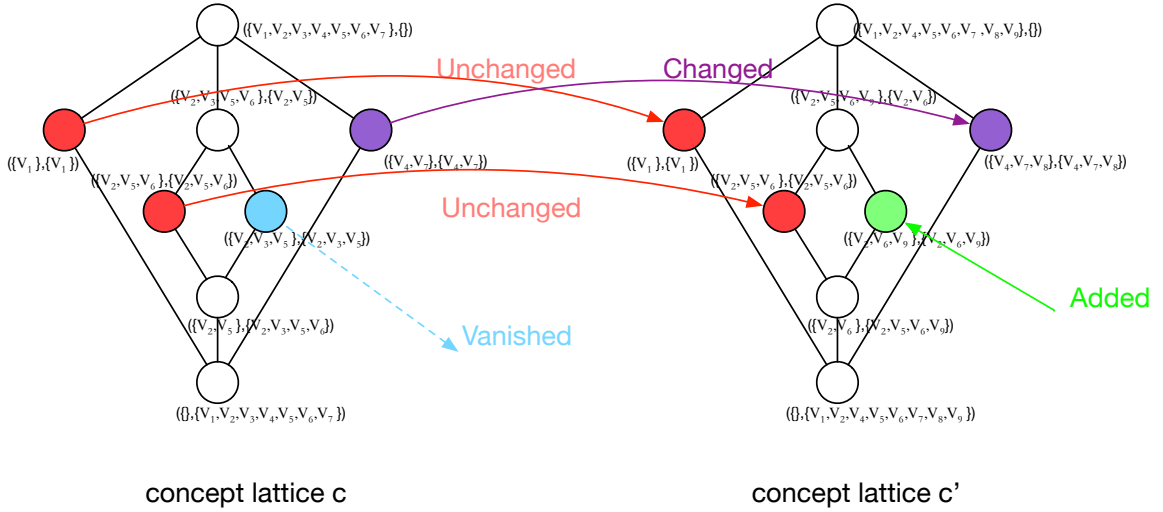


Fig. 11. The Concept Lattices of a Social Network g at Different Time.

procedure is represented with a green node and a green solid line.

- **Observation 4:** c and c' both contain $(\{1\}, \{1\})$ and $(\{2, 5, 6\}, \{2, 5, 6\})$, which are represented as red nodes and solid lines. Obviously, they are not changed.

Therefore, we can categorize equiconcepts after adding nodes as follows: (1) *unchanged equiconcept*, denoted as UE ; (2) *changed equiconcept*, denoted as CE ; (3) *added equiconcept*, denoted as AE . The equiconcepts after removing nodes can also be divided into 3 categories: (1) *unchanged equiconcept*, denoted as UE ; (2) *changed equiconcept*, denoted as CE ; (3) *vanished equiconcept*, denoted as VE .

In OSNs, we can describe the evolution of OSNs by analyzing the changes of several different categories of equiconcepts, when the nodes change. For the above example, when the social network g changes, the evolution patterns of maximal cliques are extracted as follows:

- 1) Unchanged maximal cliques: $\{1\} \rightarrow \{1\}$
 $\{2, 5, 6\} \rightarrow \{2, 5, 6\}$
- 2) Changed maximal cliques: $\{4, 7\} \rightarrow \{4, 7, 8\}$
- 3) Added maximal cliques: $\emptyset \rightarrow \{2, 6, 9\}$
- 4) Vanished maximal cliques: $\{2, 3, 5\} \rightarrow \emptyset$

We can restore the evolution of maximal cliques in OSNs by comparing equiconcepts before and after the change. This is the advantage of our proposed approach which can benefit MCP.

VII. EXPERIMENT AND ANALYSIS

This section mainly carries out experiments and evaluates the performance of the proposed approach.

A. Data Sets

We use three real-world datasets in our experiments. The description of the datasets are as follows.

- *Data set I* is a typical social network among 34 members of a Karate club at a university in the United States in the 1970s.

- *Data set II* is another dataset on the social network of frequent interactions between 62 dolphins in a community living off Doubtful sound, New Zealand [38].
- *Data set III* is collected from The Red Hot Jazz Archive digital database which is a network of 198 Jazz musicians.

B. Experiment Configuration

All experiments are implemented with Inter(R) Core(TM) i5-7500CPU@3.40GHz 3.41GHz8GB-RAM PC under Windows 10 operating system.

Since we are using the public social network datasets and the nodes in datasets are static, our experiments simulate the addition and removal of nodes in the datasets by dividing them into two parts.

- (1) If new nodes (i.e., new users) join the original social network, the nodes of the three datasets are divided into the original nodes and the added nodes. The comparison is conducted before and after adding new nodes. The evolution of maximal cliques verifies the correctness of the proposed **Add-FCA** algorithm. Consequently, the experimental datasets for Add-FCA are listed in Table III.

TABLE III
EXPERIMENTAL DATASETS FOR ADD-FCA.

Dataset	# of original nodes	# of added nodes	# of final nodes
Karate	26	8	34
	30	4	34
	32	2	34
Dolphin	42	20	62
	50	12	62
	58	4	62
Jazz	140	58	198
	180	18	198
	190	8	198

- (2) If some nodes (i.e., current users) leave from the original social network, we split the nodes of the

above three datasets into the original nodes and the removed nodes. The evolution of maximal cliques verifies the correctness of the proposed **Dec-FCA** algorithm. Consequently, the experimental datasets for Dec-FCA are listed in Table IV.

TABLE IV
EXPERIMENTAL DATASETS FOR DEC-FCA.

Dataset	# of original nodes	# of removed nodes	# of final nodes
Karate	34	10	24
	34	6	28
	34	1	33
Dolphin	62	18	44
	62	8	54
	62	2	60
Jazz	198	30	158
	198	12	186
	198	2	196

C. Experimental Results

1) *Add-FCA based Maximal Cliques Detection and Evolution*: According to Table III, the detection results of maximal cliques and their evolution are shown in Tables V, VI, and VII.

TABLE V
EXPERIMENTAL RESULTS FOR DATA SET I (KARATE) UNDER ADD-FCA.

# of original nodes	# of $\beta(K_1)$	# of added nodes	# of $\beta(K)$	# of UE	# of CE	# of AE
26	21	8	36	15	6	15
30	27	4	36	18	9	9
32	31	2	36	21	10	5

TABLE VI
EXPERIMENTAL RESULTS FOR DATA SET II (DOLPHIN) UNDER ADD-FCA.

# of original nodes	# of $\beta(K_1)$	# of added nodes	# of $\beta(K)$	# of UE	# of CE	# of AE
42	46	20	84	31	15	38
50	60	12	84	48	12	24
58	76	4	84	75	1	8

TABLE VII
EXPERIMENTAL RESULTS FOR DATA SET III (JAZZ) UNDER ADD-FCA.

# of original nodes	# of $\beta(K_1)$	# of added nodes	# of $\beta(K)$	# of UE	# of CE	# of AE
140	267	58	746	35	232	479
180	551	18	746	289	262	195
190	590	8	746	329	261	156

2) *Dec-FCA based Maximal Cliques Detection and Evolution*: According to Table IV, the detection results of maximal cliques and their evolution are shown in Tables VIII, IX, and X.

TABLE VIII
EXPERIMENTAL RESULTS FOR DATA SET I (KARATE) UNDER DEC-FCA.

# of original nodes	# of $\beta(K_1)$	# of left nodes	# of $\beta(K)$	# of UE	# of CE	# of VE
34	36	10	21	17	4	15
34	36	6	29	26	3	7
34	36	1	35	35	0	1

TABLE IX
EXPERIMENTAL RESULTS FOR DATA SET II (DOLPHIN) UNDER DEC-FCA.

# of original nodes	# of $\beta(K_1)$	# of left nodes	# of $\beta(K)$	# of UE	# of CE	# of VE
62	84	18	50	35	15	34
62	84	8	62	53	9	22
62	84	2	77	75	2	7

3) *Results Analysis*: From experimental results shown from Tables V-VII, we obtain the following relationship between the number of three types of equiconcepts when nodes are added, which is presented in Theorem 3.

Theorem 3: The number of equiconcepts generated from formal context K constructed from a social network g' equals to the summation of the number of equiconcepts generated from original formal context K_1 constructed from a social network g and the number of newly added equiconcepts under $K=K_1 \cup K_2$. It is formally represented as

$$|\beta(K)| = |\beta(K_1)| + |AE| = |UE| + |CE| + |AE|$$

where $|x|$ is the cardinality of set x .

Example 4: As can be seen from the second row of Table VII, the number of original users in Jazz social network is 180, and 551 equiconcepts are detected by the proposed Add-FCA algorithm. After adding 18 nodes to this OSN, we detect 746 equiconcepts. Comparing the equiconcepts before and after adding the nodes, we find that 289 UEs, 262 CEs, and 195 new equiconcepts have been added, and they satisfy the equation: $746 = 551 + 195 = 289 + 262 + 195$.

From the experimental results shown in Tables VIII-X, we also obtain the relationship between the number of three types of equiconcepts when nodes are removed, and this relationship can be characterized in Theorem 4.

Theorem 4: The number of equiconcepts generated from formal context K constructed from a social network g' equals the number of equiconcepts generated from the original formal context K_1 constructed from a social network g subtracting the number of vanished equiconcepts under $K=K_1-K_2$. It is formally represented as

$$|\beta(K)| = |\beta(K_1)| - |VE| = |UE| + |CE|$$

where $|x|$ is the cardinality of set x .

Example 5: Let us observe the data in the second row of Table X, the number of the original users in Jazz social network is 198, and 746 equiconcepts are detected by the Algorithm 2 BasicConcept. After 12 nodes are removed from this OSN, we detect 605 equiconcepts. Comparing the equiconcepts before

TABLE X
EXPERIMENTAL RESULTS FOR DATA SET III (JAZZ) UNDER DEC-FCA.

# of original nodes	# of $\beta(K_1)$	# of left nodes	# of $\beta(K)$	# of UE	# of CE	# of VE
198	746	30	471	125	346	275
198	746	12	605	434	171	141
198	746	2	743	737	6	3

and after removing the nodes, we find that there are 434 UEs, 171 CEs, and 141 VEs, and they meet the equation: $605 = 746 - 141 = 434 + 171$.

Comparing all the experiments, we find that the increase or decrease of nodes will lead to uncertain effects on the topology of maximal cliques in social networks. But, there are definite relationships among the number of equiconcepts obtained from formal context K and K_1 , and the evolution of maximal cliques in OSNs can be captured by investigating four types of equiconcepts (UE, CE, AE, VE).

Additionally, the solution proposed in this paper can correctly find the maximal cliques, and classify them, while reducing computational complexity under the dynamic changes of OSNs.

D. Illustrative Example

In order to better reveal the working process of our maximal cliques evolution algorithm, this section presents an illustration by utilizing a real dataset with a time span of 9 years. The dataset records the interaction between more than 1,000 directors of 384 companies in Norway from 2002 to 2011 [40]. During this period, some companies went bankrupt, and some directors switched jobs or resigned. Each director has a unique identifier (number range from 1-6000). In our paper, we use the data in August of each year as the formal context to detect and compare the maximal cliques to get the evolution of the maximal cliques each year. Further, the collaborations among these companies' directors can be easily exploited.

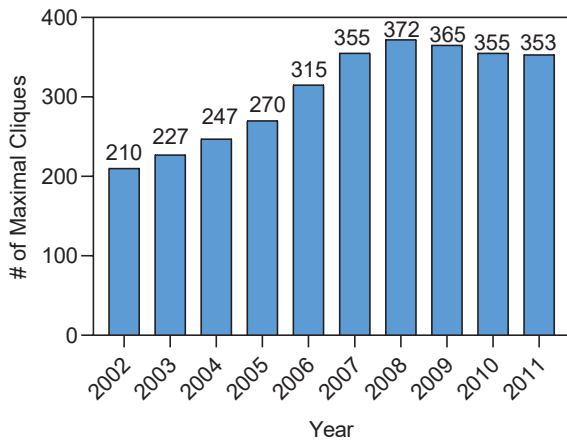


Fig. 12. The Number of Maximal Cliques from 2002 to 2011.

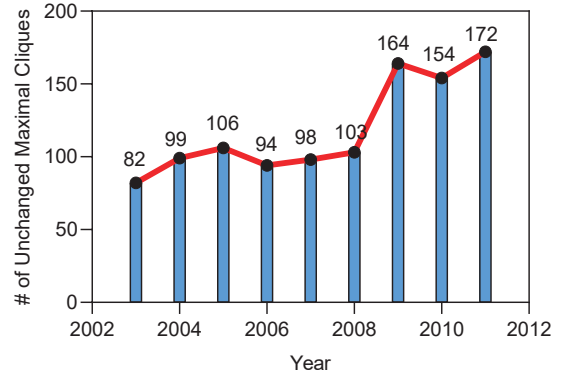


Fig. 13. The Number of Unchanged Maximal Cliques Compared with Last Year.

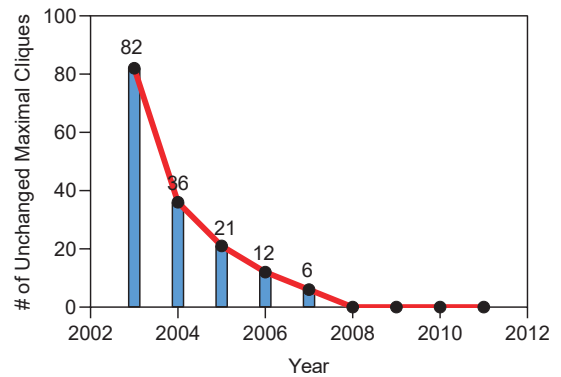


Fig. 14. The Number of Unchanged Maximal Cliques (2003-2011) vs. (2002).

Fig. 12 shows the number of maximal cliques between August 2002 and August 2011. It can be seen from this graph that the number of maximal cliques changes every year, but in general, over time, although 17 companies have closed down in 2009, the number still increased compared with that in 2002.

Fig. 13 shows the number of unchanged maximal cliques in each year from August 2002 to August 2011 (the others are changed maximal cliques, vanished maximal cliques, and added maximal). It can be seen from Fig. 13 that since 2009, the number of unchanged maximal cliques has increased, which also shows that the community structure of this social network is more stable.

Fig. 14 shows the number of unchanged maximal cliques from August 2003 to August 2011 compared with August 2002. We can see that the number is getting less and less, and by 2008 there is no longer the unchanged maximal cliques.

We specifically analyze the experimental results data corresponding to Figure 3. In 2007, there are 6 maximal cliques that are the same as in 2002: $\{204, 2193, 2258, 3521, 5082\}$, $\{2626, 3850, 4579, 5235\}$, $\{396, 3102, 3198, 3827\}$, $\{2563, 3964, 4363, 4600\}$, $\{23, 2995, 3253\}$, $\{797, 3394, 3827, 4072\}$, and in 2008, these 6 maximal cliques which exist in 2007 has either changed or vanished. Here we illustrate the first maximal clique, which

has evolved into $\{204, 1865, 2193, 2258, 5082, 5468, 5493\}$ in 2008. Compared with the formal context in 2002 and 2008, the node with ID 3521 was deleted in 2008 (It exists in the formal context in 2002), and the node with ID 1865, 5468, and 5496 were added in 2008 (It does not exist in the formal context in 2002), therefore, we regard node with ID 3521 as a key node that maintained the maximal clique $\{204, 2193, 2258, 3521, 5082\}$ from 2002 to 2007.

VIII. CONCLUSIONS

Aiming to achieve the efficient services provision in SIoT, this paper has proposed innovative solutions for efficient detection of maximal cliques and effective management of their evolution in OSN based on the FCA theory. Although the equivalence between maximal cliques and equiconcepts has been proposed and proved in our previous work, it has not considered the characteristics of dynamic changes of maximal cliques in OSNs. In addition, the evolution of maximal cliques has not been captured in the event that users are added and/or removed. This paper has proposed two algorithms, **Add-FCA** and **Dec-FCA**, that can be applied to OSNs for facilitating the generation of formal concepts due to the reconstruction of formal contexts in OSNs. The correctness and effectiveness of the proposed algorithms have been validated through extensive experiments. By capturing the evolutionary process of maximal cliques in OSNs, the proposed algorithms can quickly and accurately identify four types of maximal cliques (i.e., unchanged, changed, added and vanished maximal cliques). Importantly, a quantitative relation among the four types of maximal cliques is presented for efficiently extracting key nodes and stable communities in OSNs. In addition, a case study is presented to verify the usability of the approach and demonstrate the applicability to other potential fields such as protein structure analysis and DNA self-assembly.

ACKNOWLEDGMENT

This work was funded in part by the National Natural Science Foundation of China (Grant No. 61702317), the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 840922, the Fundamental Research Funds for the Central Universities (GK202103080). This work reflects only the authors' view and the EU Commission is not responsible for any use that may be made of the information it contains.

REFERENCES

- [1] L. Tan and N. Wang, "Future internet: The internet of things," in *2010 3rd international conference on advanced computer theory and engineering (ICACTE)*, vol. 5. IEEE, 2010, pp. V5–376.
- [2] D. Guinard, M. Fischer, and V. Trifa, "Sharing using social networks in a composable web of things," in *2010 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*. IEEE, 2010, pp. 702–707.
- [3] T.-Y. Chung, I. Mashal, O. Alsaryrah, V. Huy, W.-H. Kuo, and D. P. Agrawal, "Social web of things: a survey," in *2013 International Conference on Parallel and Distributed Systems*. IEEE, 2013, pp. 570–575.
- [4] C. L. Stergiou, K. E. Psannis, and B. B. Gupta, "Tot-based big data secure management in the fog over a 6g wireless network," *IEEE Internet of Things Journal*, 2020.
- [5] Z. Song, Y. Sun, J. Wan, L. Huang, Y. Xu, and C.-H. Hsu, "Exploring robustness management of social internet of things for customization manufacturing," *Future Generation Computer Systems*, vol. 92, pp. 846–856, 2019.
- [6] F. Hao, Z. Pei, and L. T. Yang, "Diversified top-k maximal clique detection in social internet of things," *Future Generation Computer Systems*, vol. 107, pp. 408–417, 2020.
- [7] Z. Zhang, J. Jing, X. Wang, K.-K. R. Choo, and B. B. Gupta, "A crowdsourcing method for online social networks security assessment based on human-centric computing," *Human-centric Computing and Information Sciences*, vol. 10, pp. 1–19, 2020.
- [8] L. Jin, Y. Chen, T. Wang, P. Hui, and A. V. Vasilakos, "Understanding user behavior in online social networks: A survey," *IEEE Communications Magazine*, vol. 51, no. 9, pp. 144–150, 2013.
- [9] P. Bellavista, D. Belli, S. Chessa, and L. Foschini, "A social-driven edge computing architecture for mobile crowd sensing management," *IEEE Communications Magazine*, vol. 57, no. 4, pp. 68–73, 2019.
- [10] S. Wang, A. Zhou, M. Yang, L. Sun, C.-H. Hsu, and F. Yang, "Service composition in cyber-physical-social systems," *IEEE Transactions on Emerging Topics in Computing*, vol. 8, no. 1, pp. 82–91, 2017.
- [11] X. Wang, X. Wang, F. Hao, G. Min, and L. Wang, "Efficient coupling diffusion of positive and negative information in online social networks," *IEEE Transactions on Network and Service Management*, vol. 16, no. 3, pp. 1226–1239, 2019.
- [12] P. Chen, Z. Liu, X. Qiao, and X. Tian, "The parallel algorithm of clique and it's application on data mining grid system," in *2009 IEEE International Conference on Network Infrastructure and Digital Content*. IEEE, 2009, pp. 340–343.
- [13] H. M. Grindley, P. J. Artymiuk, D. W. Rice, and P. Willett, "Identification of tertiary structure resemblance in proteins using a maximal common subgraph isomorphism algorithm," *Journal of molecular biology*, vol. 229, no. 3, pp. 707–721, 1993.
- [14] I. Koch, T. Lengauer, and E. Wanke, "An algorithm for finding maximal common subtopologies in a set of protein structures," *Journal of computational biology*, vol. 3, no. 2, pp. 289–306, 1996.
- [15] I. Koch, "Enumerating all connected maximal common subgraphs in two graphs," *Theoretical Computer Science*, vol. 250, no. 1–2, pp. 1–30, 2001.
- [16] R. Samudrala and J. Moult, "A graph-theoretic algorithm for comparative modeling of protein structure," *Journal of molecular biology*, vol. 279, no. 1, pp. 287–302, 1998.
- [17] E. J. Gardiner, P. Willett, and P. J. Artymiuk, "Graph-theoretic techniques for macromolecular docking," *Journal of Chemical Information and Computer Sciences*, vol. 40, no. 2, pp. 273–279, 2000.
- [18] R. Kumar, R. K. Behera, A. Kesarwani, and S. K. Rath, "A fast algorithm for enumerating maximal cliques in large scale network," in *2017 14th IEEE India Council International Conference (INDICON)*. IEEE, 2017, pp. 1–6.
- [19] L. Pan and E. E. Santos, "An anytime-anywhere approach for maximal clique enumeration in social network analysis," in *2008 IEEE International Conference on Systems, Man and Cybernetics*. IEEE, 2008, pp. 3529–3535.
- [20] B. Ganter and R. Wille, *Formal concept analysis: mathematical foundations*. Springer Science & Business Media, 2012.
- [21] Z. Chen, "New algorithm of generating concept lattice based on concept-matrix," *Computer Science*, 2010.
- [22] L. Nourine and O. Raynaud, "A fast incremental algorithm for building lattices," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 14, no. 2-3, pp. 217–227, 2002.
- [23] M. Chein, "Algorithme de recherche des sous-matrices premières d'une matrice," *Bulletin mathématique de la Société des Sciences Mathématiques de la République Socialiste de Roumanie*, pp. 21–25, 1969.
- [24] J. Ma and L. Hu, "Attribute reductions of interval-set concept lattices for decision formal contexts," *Pattern Recognition and Artificial Intelligence*, vol. 31, no. 7, pp. 581–590, 2018.
- [25] T. Qian and X. He, "The construction theoretical research of interval-set concept lattices," *Journal of Northwest University(Natural Science Edition)*, vol. 47, no. 3, pp. 330–335, 2017.
- [26] X. He, L. Wei, and T. Qian, "Property oriented interval-set concept lattice," *Journal of Frontier of Computer Science and Technology*, vol. 12, no. 9, pp. 150–156, 2018.
- [27] L. Zeng, Z. Zhang, and L. Zou, "Fast algorithm for incremental construction of concept lattice," *Journal of Frontier of Computer Science*, vol. 12, no. 11, pp. 1862–1870, 2018.

- [28] L. Zhang, H. Zhang, L. Yin, and D. Han, "Theory and algorithms of attribute decrement for concept lattice," *Journal of Computer Research and Development*, vol. 50, no. 2, p. 248, 2013.
- [29] Y. Wan and L. Zou, "An efficient algorithm for decreasing the granularity levels of attributes in formal concept analysis," *IEEE Access*, vol. 7, pp. 11 029–11 040, 2019.
- [30] L. Zou, Z. Zhang, and J. Long, "An efficient algorithm for increasing the granularity levels of attributes in formal concept analysis," *Expert Systems with Applications*, vol. 46, pp. 224–235, 2016.
- [31] Y. Yang, F. Hao, B. Pang, K. Qin, Z. Pei, and B. Li, "A quick algorithm on generating concept lattice for attribute-incremental streaming data," in *2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*. IEEE, 2019, pp. 2811–2816.
- [32] A. Das, S.-V. Sanei-Mehri, and S. Tirthapura, "Shared-memory parallel maximal clique enumeration," in *2018 IEEE 25th International Conference on High Performance Computing (HiPC)*. IEEE, 2018, pp. 62–71.
- [33] J. Cheng, L. Zhu, Y. Ke, and S. Chu, "Fast algorithms for maximal clique enumeration with limited memory," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2012, pp. 1240–1248.
- [34] D. Bera, M. Pal, and T. Pal, "An efficient algorithm to generate all maximal cliques on trapezoid graphs," *International journal of computer mathematics*, vol. 79, no. 10, pp. 1057–1065, 2002.
- [35] V. Snášel, Z. Horák, J. Kocibova, and A. Abraham, "Reducing social network dimensions using matrix factorization methods," in *2009 International Conference on Advances in Social Network Analysis and Mining*. IEEE, 2009, pp. 348–351.
- [36] V. Snášel, Z. Horák, J. Kocibova, and A. Abraham, "Analyzing social networks using fca: complexity aspects," in *2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, vol. 3. IEEE, 2009, pp. 38–41.
- [37] S. Krajčič, "Social network and formal concept analysis," in *Social Networks: A Framework of Computational Intelligence*. Springer, 2014, pp. 41–61.
- [38] F. Hao, G. Min, Z. Pei, D.-S. Park, and L. T. Yang, " k -clique community detection in social networks based on formal concept analysis," *IEEE Systems Journal*, vol. 11, no. 1, pp. 250–259, 2017.
- [39] F. Hao, D.-S. Park, and Z. Pei, "When social computing meets soft computing: opportunities and insights," *Human-centric Computing and Information Sciences*, vol. 8, no. 1, pp. 1–18, 2018.
- [40] C. Seierstad and T. Opsahl, "For the few not the many? the effects of affirmative action on presence, prominence, and social capital of women directors in norway," *Scandinavian journal of management*, vol. 27, no. 1, pp. 44–54, 2011.



Fei Hao received the B.Sc. degree in Information and Computing Science and the M.Sc. degree in Computer Software and Theory from Xihua University, China, in 2005 and 2008, respectively, and the Ph.D. degree in Computer Science and Engineering from Soonchunhyang University, South Korea, in 2016. Since 2016, he has been with Shaanxi Normal University, Xi'an, China, where he is an Associate Professor. He is currently taking a Marie Skłodowska-Curie Individual Fellowship at the University of Exeter, Exeter, United Kingdom.

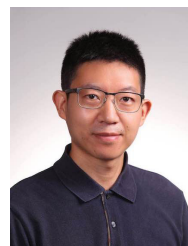
His research interests include social computing, ubiquitous computing, big data analysis and processing and mobile cloud computing.



Beibei Pang received the B.M. degree in Information Management and Information Systems from Shaanxi Normal University, China, in 2017. She is currently working toward the M.Sc. degree at the School of Computer Science, Shaanxi Normal University, China. Her current research interests include social computing, mobile edge computing and big data.



Geyong Min is a Professor of High Performance Computing and Networking in the Department of Computer Science within the College of Engineering, Mathematics and Physical Sciences at the University of Exeter, United Kingdom. He received the PhD degree in Computing Science from the University of Glasgow, United Kingdom, in 2003, and the B.Sc. degree in Computer Science from Huazhong University of Science and Technology, China, in 1995. His research interests include Computer Networks, Wireless Communications, Parallel and Distributed Computing, Ubiquitous Computing, Multimedia Systems, Modelling and Performance Engineering.



Yulei Wu is a Senior Lecturer with the Department of Computer Science, College of Engineering, Mathematics and Physical Sciences, University of Exeter, United Kingdom. He received the B.Sc. degree (First Class Honours) in Computer Science and the Ph.D. degree in Computing and Mathematics from the University of Bradford, United Kingdom, in 2006 and 2010, respectively. His expertise is on intelligent networking, and his main research interests include computer networks, networked systems, software defined networks and systems, network management, and network security and privacy. He is an Editor of IEEE Transactions on Network and Service Management, IEEE Transactions on Network Science and Engineering, Computer Networks (Elsevier) and IEEE Access. He is a Senior Member of the IEEE and the ACM, and a Fellow of the HEA (Higher Education Academy).



Yixuan Yang received the M.Sc. degree in Software Engineering from Shaanxi Normal University, China and the B.Sc. degree in Software Engineering from Taiyuan University of Technology, China, in 2020 and 2017, respectively. Her research interests include social computing and formal concept analysis.