Dynamic and Overlapping Community Detection in Link Streams through Formal Concept Analysis

Martin WAFFO KEMGNE[1], Christophe DEMKO[1], Jean-Loup GUILLAUME[1], and Karell BERTET[1]

L3i (Laboratoire Informatique, Image et Interaction),
La Rochelle Université, La Rochelle, France
{martin.waffo_kemgne, christophe.demko, jean-loup.guillaume, karell.bertet}@univ-lr.fr

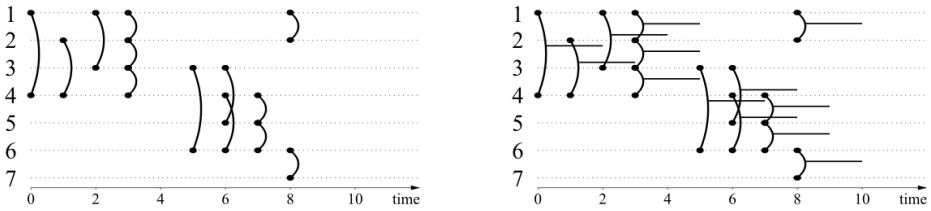
Abstract. Community detection is central to network analysis, but most methods target static structures. Real-world communities are often overlapping and dynamic, requiring more precise models. While methods like Clique Percolation address overlap, they typically rely on coarse time discretizations that obscure temporal patterns. The link stream (LS) model captures time-stamped interactions, with LSCPM being the only known method for overlapping community detection in this setting. In this paper, we adapt a Formal Concept Analysis-based method to the LS context, enabling high-resolution detection of dynamic, overlapping communities. We also propose tailored evaluation metrics for this temporal context.

Keywords: Dynamic networks, Community detection, Formal Concept Analysis, Link Streams

1 Introduction

The proliferation of large-scale, complex data has increased the importance of community detection, a key task in network science with many applications in social network analysis and mining [7]. Overlapping community structures, where nodes belong to multiple groups, are common in real-world networks but challenge traditional methods based on disjoint partitions. Among other solutions, the Clique Percolation Method (CPM) defines communities as unions of overlapping cliques [16]. Despite its utility, CPM and similar methods are sensitive to parameter choices, computationally demanding on large networks, and can be less effective in dynamic settings.

While community detection has also been extended to dynamic networks using models such as Snapshot Graphs, Temporal Networks, and Time-Varying Graphs [11,17], these approaches often rely on arbitrary time discretizations. For instance, the **snapshot model** represents a network as a sequence of static graphs and, despite its simplicity, this model relies on arbitrary time slicing. The Link Stream (LS) model overcomes this by capturing interactions as they occur [13]. While LS has been used in temporal pattern mining and social dynamics,



(a) Instantaneous link stream. (b) Interval-based link stream.

Fig. 1: Representation of a link stream with instantaneous links (a) and with intervals (b). The duration of interval links is represented with horizontal lines. Note that the interval-based LS is the 2-transformation of the instantaneous one.

community detection remains largely underexplored. In particular, LSCPM [3], the only extension of CPM to LS, is parametric and sensitive to clique size.

To address these limitations, we propose a hierarchical non-parametric method based on Formal Concept Analysis (FCA) for detecting overlapping communities in LS. FCA has inspired several static community detection methods [6,8,12]. Our method uses maximal cliques in LS to build hierarchical and overlapping communities without parameter tuning. It constructs a concept lattice directly from the data, and offers robust, interpretable, and adaptable community detection. To enable evaluation in LS, we also propose adapted versions of modularity [15] and F1-score [4], addressing the current lack of suitable metrics.

2 Preliminary Definitions

2.1 Links Streams

Link streams (LS) offer a framework for modeling time-resolved interactions [13]. There are two main variants **interval-based** and **instantaneous** link streams (Fig. 1) to model interactions respectively with or without duration (e.g., phone calls vs email exchanges). We mostly follow the definitions from [13].

Definition 1 (Interval-Based Link Stream). *An interval-based LS is a triplet $L = (T, V, E)$, where T is a time interval, V is a set of vertices, and $E \subseteq T \times T \times V \times V$ is a set of links (b, e, u, v) such that $e \geq b$. Each link represents an interaction between vertices u and v that persists from time b to time e. The duration of the link is defined as $e - b$.*

Definition 2 (Instantaneous Link Stream). *A LS with instantaneous links is a triplet $L = (T, V, E)$, where T is a time interval, V is a set of vertices, and $E \subseteq T \times V \times V$ is a set of instantaneous links (t, u, v), which signifies that vertices u and v interact at time t, without any duration.*

Definition 3 (Δ-Transformation). *The Δ-transformation converts an instantaneous link (t, u, v) into a interval link $(t, t + \Delta, u, v)$. Given a link stream $L = (T, V, E)$, this transformation is applied to each instantaneous link in L.*

Detecting communities in a LS $L = (T, V, E)$ can be viewed as clustering the set of temporal nodes defined as follows:

Definition 4 (Temporal node). *In an interval-based LS $L = (T, V, E)$, a temporal node is a pair (u, t), where $u \in V$, $t \in T$, and such that there exists a link $(b, e, u, v) \in E$ with $t \in [b, e]$, i.e. u is connected at time t.*

A community in $L = (T, V, E)$ is a set of temporal nodes $\{(u, I_u)\} \subseteq V \times T$, where I_u is a set of disjoint intervals during which u belongs to the community. A key notion for community detection in graphs is the *clique* which is a subset of nodes where every pair of nodes is connected by an edge. This definition naturally extends to LS, by simultaneously considering time and structure.

Definition 5 (Clique, Maximal Clique and k-Clique). *A clique in a LS is a pair $(C, [t_0, t_1])$, where $C \subseteq V$, $|C| \geq 2$, and $t_0 < t_1$, such that for all $u \neq v \in C$, there exists a link (b, e, u, v) with $[t_0, t_1] \subseteq [b, e]$. It is* maximal *if it cannot be extended in time or node set without losing this property. If $|C| = k$, it is a k-clique; a maximal k-clique is a time-maximal clique of size k.*

Notations: Throughout the paper, LS denotes an interval-based LS without self-loops (i.e., $u \neq v$ for any link $(b, e, u, v) \in E$). Multiple links between the same pair of vertices must occur over disjoint intervals: if (b, e, u, v) and $(b', e', u, v) \in E$, then $[b, e] \cap [b', e'] = \emptyset$. We denote a clique in an LS by **CLS**, a maximal clique by **MCLS**, and a maximal k-clique by **MkCLS**.

2.2 Formal Concept Analysis

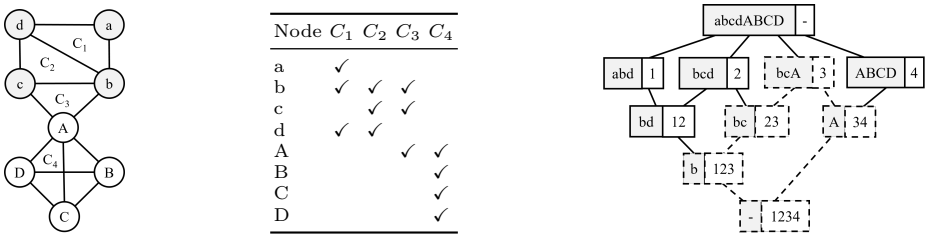
FCA is a data analysis method based on lattice theory, used to identify hierarchical relationships between objects and their attributes [9].

Definition 6 (Formal Context). *A formal context $\mathbb{K} = (G, M, I)$ consists of a set of objects G, a set of attributes M, and an incidence relation $I \subseteq G \times M$ that indicates which objects are linked to which attributes.*

Definition 7 (Galois Connection Operators). *In FCA, the operators $\alpha : 2^G \to 2^M$ and $\beta : 2^M \to 2^G$ define a Galois connection and are defined by:*
- $\alpha(A) = \{m \in M \mid \forall g \in A, (g, m) \in I\}$ *for $A \subseteq G$,*
- $\beta(B) = \{g \in G \mid \forall m \in B, (g, m) \in I\}$ *for $B \subseteq M$.*

Definition 8 (Formal Concept and Concept Lattice). *A formal concept is a pair (A, B), where $A \subseteq G$ and $B \subseteq M$, satisfying $\alpha(A) = B$ and $\beta(B) = A$. Here, A is called the extension and B the intension. The set of all formal concepts forms a complete lattice, called the concept lattice, ordered by $(A_1, B_1) \geq (A_2, B_2) \Leftrightarrow A_1 \supseteq A_2 \Leftrightarrow B_2 \supseteq B_1$.*

This lattice defines a hierarchy of concepts, where the top element includes all objects and the bottom element includes all attributes. Figure 2b illustrates



Node	C_1	C_2	C_3	C_4
a	✓			
b	✓	✓	✓	
c		✓	✓	
d	✓	✓		
A			✓	✓
B				✓
C				✓
D				✓

(a) Graph (b) Formal context (c) Concept lattice

Fig. 2: From Graphs to Communities: (a) a simple graph with 4 cliques. (b) the formal context that identifies the nodes and the cliques they belong to. (c) the concept lattice that captures all the relevant intersections of cliques.

the formal context where objects are nodes and attributes are the cliques to which they belong. The corresponding concept lattice is given in Figure 2c.

Some formal concepts are of particular interest, namely the *object concepts* and the *attribute concepts*. In Figure 2c, the concept `b|123` is the first (from the bottom) where node `b` appears, it is called an object concept. Similarly, `bcd|2` is an attribute concept (the first from the top where clique `2` appears).

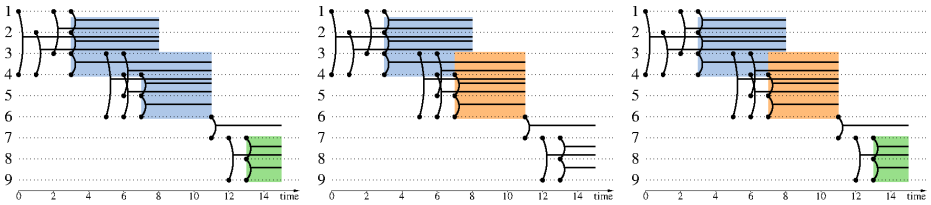
Definition 9 (Object Concept and Attribute Concept). *An* object concept *is a formal concept (A, B) that introduces a new object $g \in G$; that is, there exists $g \in G$ such that $g \in A$ but $g \notin A'$ for all formal concept (A', B'), such that $(A, B) \geq (A', B')$ and $(A, B) \neq (A', B')$. Symmetrically, one can define attribute concepts based on attributes instead of objects.*

3 Related Work

Community detection has been widely explored, spanning static and dynamic networks [7]. Existing methods fall into two main categories: non-overlapping and overlapping approaches. Overlapping methods allow nodes in multiple communities and metrics like modularity have been adapted to this setting [18]. Other approaches include line graphs for edge clustering [5] and Clique Percolation Method (CPM) [16], which links k-cliques that share $k-1$ nodes.

Community Detection Based on FCA. Given a graph $G = (V, E)$, let $\mathcal{F} = \{F_1, \ldots, F_l\}$, $F_i \subset V$, be a set of *atomic communities* (e.g., cliques) that can be combined to form larger communities. These are grouped into clusters $\pi = \{P_1, \ldots, P_m\}$ using an equivalence relation such that $P_i \cap P_j = \emptyset$ for $i \neq j$, and $\bigcup_i P_i = \mathcal{F}$. This generic framework applies to several methods, including CPM where the k-cliques are atomic communities that form the communities once merged when they share $k - 1$ nodes.

In FCA-based approaches, atomic communities are maximal cliques. Freeman [8] pioneered the use of FCA in community detection but only using overlap of maximal cliques on the first layer and also excluding bridging cliques.



(a) LSCPM ($k = 3$). (b) LSCPM ($k = 4$). (c) Our approach.

Fig. 3: Comparison of LSCPM and our approach on a LS with three communities.

Falzon [6] added bridging cliques and applied the overlap relation across all lattice layers. This approach is extended in FALO [12] to detect overlapping communities and automatically select the best decomposition.

Communities in evolving networks. Despite numerous models for dynamic networks [11,13], and several associated algorithms, few use the LS formalism. Most approaches struggle with real-world scenarios where the network is not consistently well-formed, especially in stream-based models. For example, when an email is sent between two people, the aggregated graph at that precise moment only reflects the interaction between these two nodes, omitting temporal context. Fixed aggregation windows either miss communities (if too short) or merge unrelated events (if too long).

Baudin et al. [3] introduced LSCPM, a generalization of CPM using MkCLS instead of static k-cliques in graphs. An LSCPM community consists of temporal vertices in a maximal set of MkCLS, connected via adjacent MkCLS, i.e., those sharing $k-1$ nodes and overlapping in time. While efficient, LSCPM is sensitive to the clique size k. Figure 3 highlights this: for $k = 3$, LSCPM merges the blue and orange communities; for $k = 4$, it misses the green one; for $k > 4$, it detects none. In constrast, our approach consistently identifies all communities (see Fig. 3c).

4 Proposed Approach

We propose an FCA-based method for detecting communities in LS inspired by FALO [12], henceforth called LSFALO. It constructs the concept lattice using maximal cliques and then extracts the communities from the lattice.

From Link Stream to Concept Lattice Given a LS $L = (T, V, E)$, we detect MCLS using Baudin's algorithm [2]. Let $C = \{C_1, \ldots, C_n\}$ be the set of MCLS in L, where each $C_i = (\tilde{C}_i, [t_0^i, t_1^i])$, with $\tilde{C}_i \subseteq V$, contains at least three nodes. Let $\tilde{G}$ denote the set of temporal nodes (t, u) (see Definition 4). We then construct the formal context $\mathbb{K} = (\tilde{G}, C, I)$, where $I((u,t), C_i) = 1$ if $(u,t) \in \tilde{C}_i \times [t_0^i, t_1^i]$, and 0 otherwise, for all $i = 1..n$.

We then build the concept lattice $\mathbb{L} = \{L_0, ..., L_k\}$. The first level L_0 consists solely of the top concept $\top = (V, I_V)$, where I_V is the maximal time interval during which all nodes in V are simultaneously connected. If no such interval exists (i.e., the entire LS is never a complete clique), then I_V is the empty set. The level L_1 gathers the sets of temporal nodes associated with individual MCLS. Higher levels L_i (for $i \geq 2$) are built by intersecting the node sets of pairs of concepts from L_{i-1}. After each intersection step, strictly included concepts are removed to retain only maximal ones. Finally, **object concepts** (OC) are extracted at each level to identify newly introduced temporal nodes.

From Formal Lattice to communities We extract communities from $\mathbb{L}$ on each layer L_k, where each concept $n_i \in L_k$ is of the form $n_i = (\tilde{K_i}, [t_0^i, t_1^i])$, where $\tilde{K_i}$ is a set of nodes and $[t_0^i, t_1^i]$ an associated time interval. We define the $o_{tc} \subseteq L_k \times L_k$ overlap relation by $(n_i, n_j) \in o_{tc} \iff |\tilde{K_i} \cap \tilde{K_j}| > c$ and $[t_0^i, t_1^i] \cap [t_0^j, t_1^j] \neq \emptyset$. This means that two concepts $n_i, n_j \in L_k$ are overlapping if they share more than c objects and their time intervals intersect. We then consider the **transitive closure** of o_{tc}, i.e., we impose that if $(n_i, n_j) \in o_{tc}$ and $(n_j, n_k) \in o_{tc}$, then $(n_i, n_k) \in o_{tc}$ as well.

Following the general principle described in Section 3, we group concepts, viewed as atomic communities, into larger temporal communities on each layer L_k of $\mathbb{L}$. The grouping uses the equivalence classes of o_{tc}. For each equivalence class T_k, we aggregate the set of all nodes G_k from the concepts in the class. For each node $u \in G_k$, we collect all time intervals during which u appears and merge them into disjoint intervals $\tilde{T_u}$, and assigns them to the corresponding node. Finally, the temporal community $\{(u, \tilde{T_u}) \mid u \in G_k\}$ is added to the list of communities at level i. The process is repeated for all levels, resulting in a hierarchy of nested temporal communities. Note that c is computed from the lattice as the minimum cardinality among non-empty object sets, the method therefore remains non-parametric and data-driven.

5 Results and Discussions

We evaluate our method on both synthetic and real-world datasets, and compare it with LSCPM [3], the only community detection algorithm for LS.

5.1 Evaluation metrics

Community quality can be assessed using intrinsic and extrinsic metrics. Among the most common are modularity [15], which compares internal versus expected edges, and F1-score [4], which evaluates accuracy against a ground truth. We extend these two definitions to account for overlapping communities in LS.

Modularity for LS. We combine overlapping-aware terms from Shen et al. [18] with the Constant Potts Model from Traag et al. [19] to obtain an LS modularity.

Given a LS $L = (T, V, E)$ and c a temporal community composed of vertices (u, I_u), where I_u is a set of disjoint intervals during which node u is present in c, the modularity Q_{LS} for a partition $\{C_1, \ldots, C_n\}$ is:
$Q_{LS} = 1 + \frac{1}{2N} \sum_{l=1}^{n} \sum_{(u,I_u),(v,I_v)\in C_l} \frac{1}{O_u O_v} \left(A_{uv}(c) - 1\right)$, where O_i is the number of communities node i belongs to, $A_{uv}(c) = \frac{\sum_{t=1}^{k} |T_t|}{\alpha(c)}$ if u and v interact over intervals $\{T_1, \ldots, T_k\}$ within c, 0 otherwise. $\alpha(c) = \max_{(u,I_u)\in c,(b,e)\in I_u} e - \min_{(v,I_v)\in c,(b,e)\in I_v} b$, is a normalization term which captures the maximal temporal span of community c (i.e., the time between the earliest appearance and latest disappearance of its nodes), preventing communities artificially stretched over long periods from being favored. The $+1$ shift ensures $Q_{LS} \in [0, 1]$.

F1-score for LS. To extend the $\tilde{F}1$-score [4] to LS, we adapt the notion of overlap to temporal communities. Let $C = \{C_i\}$ and $C' = \{C'_j\}$ be sets of ground-truth and detected temporal communities. Each C_i is composed of temporal vertices (u, I_u), where u is a node and I_u is a set of disjoint intervals. The temporal size of a community is $|C_i|_T = \sum_{(u,I_u)\in C_i} \sum_{I\in I_u} |I|$, and their temporal intersection is $|C_i \cap C'_j|_T = \sum_{(u,I_u)\in C_i,(u,J_u)\in C'_j} \sum_{I\in I_u, J\in J_u} |I \cap J|$.. This allows us to define precision, recall and F1-score between two communities. Similar to [4], the $\tilde{F}1_{LS}(C, C')$ searches the best match in C' for each community in C and vice-versa.

5.2 Datasets

Synthetic Data. Since real-world dynamic networks often lack ground truth, we constructed synthetic networks with predefined community structures using the Mosaic benchmark [1]. This model generates LS by providing fine-grained control over key parameters: number of nodes, maximum time interval, internal and external connection probabilities, Poisson-based interaction frequencies, and the probability of empty Mosaic communities.

Real-World Data. We also test our method on publicly available real-world LS datasets[1]. **HSD11–13** are High school contact networks collected in Marseille, France, across 2011–2013 [14], **WC1–2** are workplace interactions in France [10]. Table 1 summarizes their key characteristics.

Contacts are measured using wearable proximity sensors, with a contact recorded when sensors remain close for a 20s duration (called time resolution). These datasets are thus interval-based LS, with interval lengths being multiples of 20 seconds.

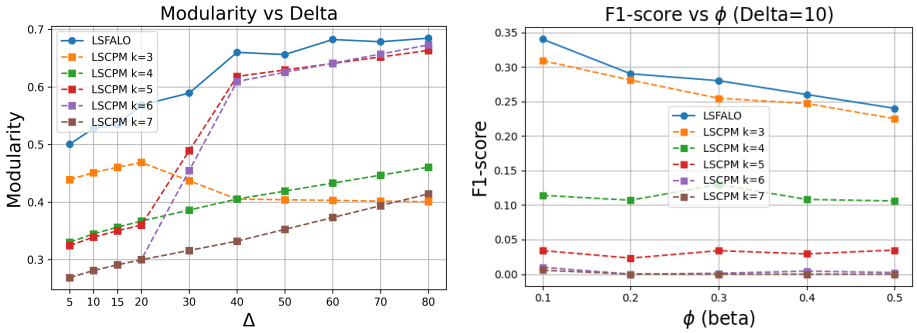
5.3 Performance Evaluation

We used the Python libraries `streamfig` and `mosaic_benchmark` for the experiments. Evaluations were performed on a desktop with a Core i9-12900H

[1] `http://www.sociopatterns.org/datasets/`

Table 1: Statistics of the real-world datasets: type of LS (with the resolution: each interval is a multiple of the resolution), number and type of nodes and temporal edges, total duration of the measurement.

Network	Type (res)	Duration	Nodes	Edges
HSD11	Interval (20s)	4 days	126 Students & Teachers	28.5k Contacts
HSD12	Interval (20s)	7 days	180 Students	45k Contacts
HSD13	Interval (20s)	5 days	327 Students	188.5k Contacts
WC1	Interval (20s)	14 days	145 Individuals	1193 Contacts
WC2	Interval (20s)	2 days	403 Individuals	9.6k Contacts



(a) Modularity vs. Δ. (b) F1-score vs. ϕ.

Fig. 4: Impact of Δ on Modularity and ϕ on F1-score (using $\Delta = 10$) with LSCPM ($k = 3 \dots 7$) and LSFALO. Mosaic generated datasets with 100 nodes and the default parameters (except ϕ for (b)).

processor at 2.50 GHz and 32 GB of RAM. Results were averaged over 10 independent runs. The source code used in our experiments is publicly available at: `https://gitlab.univ-lr.fr/mwaffo01/lsfalo.git`.

Evaluations on Synthetic Networks. We used the Mosaic benchmark, using parameters similar to those of the authors. Each network had $n = 100$ nodes interacting over time. We performed two sets of simulations based on the time window Δ and the community distinguishability parameter ϕ of the Mosaic benchmark.

We first analyzed the impact of Δ on community quality using modularity (Figure 4a). Modularity increases with Δ as larger cliques emerge, and *LSFALO* consistently outperforming all *LSCPM* variants. Beyond a certain Δ, modularity plateaus, indicating diminishing returns. In contrast, *LSCPM* is sensitive to its parameter k: $k = 3$ performs well for small Δ, but degrades with larger cliques requiring different k values. No single k is optimal, highlighting *LSFALO*'s robustness.

Table 2: Execution time (in seconds) for LSCPM ($k = 3$ and $k = 7$) and LSFALO with different values of Δ (20 seconds, 10 minutes, 1 hour) on five real datasets.

Method	HSD11			HSD12			HSD13			WC1			WC2		
Δ	20s	10mn	1h	20s	10mn	1h	20s	10mn	1h	20s	10mn	1h	20s	10mn	1h
LSCPM-3	0.07	0.08	0.09	0.10	0.09	0.08	0.13	0.14	0.14	0.08	0.06	0.07	0.13	0.13	0.30
LSCPM-7	0.07	0.11	0.11	0.09	0.10	0.09	0.14	0.15	0.15	0.07	0.06	0.08	0.13	0.12	0.18
LSFALO	0.09	55.4	154	0.16	0.19	0.24	0.17	1.69	17.13	0.13	0.14	0.13	0.17	16.23	2541

We also examined the effect of the community distinguishability parameter ϕ in the Mosaic benchmark. Higher ϕ values reduce internal links, making detection harder. Varying ϕ from 0.1 to 0.5, consistently achieved higher F1-scores (Figure 4b). Both methods decline with increasing ϕ, though *LSFALO* remains superior. Modularity results, omitted for space, show similar trends.

Scalability on real world networks. *LSCPM* is generally faster due to fixed-size clique enumeration, while *LSFALO* incurs extra cost due from lattice generation and exploration (Table 2). However, for small time windows ($\Delta = 20s$) the overhead is limited, and runtimes are comparable. Even with larger Δ, *LSFALO* often achieves its best modularity in under a second (HSD12, WC1 for $\Delta = 1h$), though computation time can increase significantly (HSD11, WC2 for $\Delta = 1h$). Overall, while runtime grows with Δ, *LSFALO* remains scalable in practical contexts, and consistently matches or exceeds *LSCPM*'s modularity.

6 Conclusion

This paper introduces the first approach to leveraging Formal Concept Analysis for community detection in link streams. Beyond its performance, it offers several advantages over LSCPM: its non-parametric nature ensures robustness across different data types, and its hierarchical structure provides a multi-level view of communities. We also extend two popular metrics, Modularity and F1-score, to enable the evaluation of community quality and performance comparisons between algorithms in link streams.

Since the Mosaic benchmark is the only tool available for generating link streams, and it produces non-overlapping streams, our ability to thoroughly test the algorithm is limited. The lack of ground truth in real-world datasets further constrains our evaluation. These challenges highlight the need for more advanced network generators and datasets with ground-truth annotations. We also plan to improve the scalability of our approach by optimizing its complexity, particularly by generating only portions of the lattice to reduce memory usage, and enhancing the community detection phase for greater efficiency.

Acknowledgments. This research work is supported by the French National Research Agency through the project "SmartFCA - ANR-21-CE23-0023: Formal Concept Analysis: an intelligent tool for the analysis of complex data."

References

1. Asgari, Y., Cazabet, R., Borgnat, P.: Mosaic benchmark networks: Modular link streams for testing dynamic community detection algorithms. In: Intl Conference on Complex Networks and Their Applications. pp. 209–222 (2023)
2. Baudin, A., Magnien, C., Tabourier, L.: Faster maximal clique enumeration in large real-world link streams. arXiv preprint arXiv:2302.00360 (2023)
3. Baudin, A., Tabourier, L., Magnien, C.: Lscpm: communities in massive real-world link streams by clique percolation method. arXiv preprint arXiv:2308.10801 (2023)
4. Epasto, A., Lattanzi, S., Paes Leme, R.: Ego-splitting framework: From non-overlapping to overlapping clusters. In: 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 145–154 (2017)
5. Evans, T.S., Lambiotte, R.: Line graphs, link partitions, and overlapping communities. Physical review E **80**(1), 016105 (2009)
6. Falzon, L.: Determining groups from the clique structure in large social networks. Social networks **22**(2), 159–172 (2000)
7. Fortunato, S.: Community detection in graphs. Physics reports **486**(3-5), 75–174 (2010)
8. Freeman, L.C.: Cliques, galois lattices, and the structure of human social groups. Social networks **18**(3), 173–187 (1996)
9. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer Cham, 2 edn. (2024). https://doi.org/10.1007/978-3-031-63422-2
10. Génois, M., Vestergaard, C.L., Fournet, J., Panisson, A., Bonmarin, I., Barrat, A.: Data on face-to-face contacts in an office building suggest a low-cost vaccination strategy based on community linkers. Network Science **3**, 326–347 (9 2015)
11. Holme, P., Saramäki, J.: Temporal networks. Physics reports **519**(3), 97–125 (2012)
12. Kemgne, M.W., Demko, C., Bertet, K., Guillaume, J.L.: Fuzzy and overlapping communities detection: An improved approach using formal concept analysis. In: 16th Intl Conference on Advances in Social Networks Analysis and Mining (2024)
13. Latapy, M., Viard, T., Magnien, C.: Stream graphs and link streams for the modeling of interactions over time. Social Network Analysis and Mining **8**, 1–29 (2018)
14. Mastrandrea, R., Fournet, J., Barrat, A.: Contact patterns in a high school: A comparison between data collected using wearable sensors, contact diaries and friendship surveys. PLOS ONE **10**(9), 1–26 (09 2015)
15. Newman, M.E., Girvan, M.: Finding and evaluating community structure in networks. Physical review E **69**(2), 026113 (2004)
16. Palla, G., Derényi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. nature **435**(7043), 814–818 (2005)
17. Rossetti, G., Cazabet, R.: Community discovery in dynamic networks: a survey. ACM computing surveys (CSUR) **51**(2), 1–37 (2018)
18. Shen, H., Cheng, X., Cai, K., Hu, M.B.: Detect overlapping and hierarchical community structure in networks. Physica A: Statistical Mechanics and its Applications **388**(8), 1706–1712 (2009)
19. Traag, V.A., Van Dooren, P., Nesterov, Y.: Narrow scope for resolution-limit-free community detection. Physical Review E—Statistical, Nonlinear, and Soft Matter Physics **84**(1), 016114 (2011)