Information Processing Letters 133 (2018) 44–48

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

Enumerating maximal cliques in link streams with durations

Tiphaine Viard [a], Clémence Magnien [b,*], Matthieu Latapy [b]

[a] CEDRIC CNAM, F-75003 Paris, France
[b] Sorbonne Universités, UPMC Univ Paris 06, UMR 7606, LIP6, F-75005, Paris, France

ARTICLE INFO

Article history:
Received 1 August 2017
Received in revised form 22 January 2018
Accepted 22 January 2018
Available online 3 February 2018
Communicated by B. Doerr

Keywords:
Link streams
Temporal networks
Time-varying graphs
Cliques
Graph algorithms

ABSTRACT

Link streams model interactions over time, and a clique in a link stream is defined as a set of nodes and a time interval such that all pairs of nodes in this set interact permanently during this time interval. This notion was introduced recently in the case where interactions are instantaneous. We generalize it to the case of interactions with durations and show that the instantaneous case actually is a particular case of the case with durations. We propose an algorithm to detect maximal cliques that improves our previous one for instantaneous link streams, and performs better than the state of the art algorithms in several cases of interest.

1. Introduction

A **graph** is a pair of sets $G = (V, E)$ where V is a set of nodes and $E \subseteq V \times V$ a set of links. If there is no distinction between (u, v) and (v, u) for u and v in V, then G is undirected. If for all (u, v) in E, $u \neq v$ then G is simple. One generally considers undirected simple graphs. A clique of G is a set of nodes $C \subseteq V$ such that all nodes of C are linked to each other, *i.e.* for all $\{u, v\} \subseteq C$, $(u, v) \in E$. A clique C is maximal if there is no other clique C' such that $C \subset C'$. Enumerating maximal cliques of a graph is one of the most fundamental problems in computer science [1,2,10,4].

An **instantaneous link stream** is a triplet $L = (T, V, E)$ where T is a time interval, V a set of nodes and $E \subseteq T \times V \times V$ a set of instantaneous links. If there is no distinction between (t, u, v) and (t, v, u), then L is undirected. If for all (t, u, v) in E, $u \neq v$ then L is simple. For any given duration Δ, we introduced in [12] the notion of **Δ-clique** in a simple undirected link stream $L = (T, V, E)$: it is a pair $C = (X, [x, y])$ with $X \subseteq V$ and $[x, y] \subseteq T$ such that for all $\{u, v\} \subseteq X$ and $I \subseteq [x, y]$ with $|I| = \Delta$ there is a t in I such that $(t, u, v) \in E$. In other words, there is an interaction between all pairs of nodes in X at least once every Δ within $[x, y]$. A clique $C = (X, [x, y])$ is maximal if there is no other clique $C' = (X', [x', y'])$ such that $C' \neq C$, $X \subseteq X'$ and $[x, y] \subseteq [x', y']$. We proposed a first algorithm to enumerate all maximal Δ-cliques of an instantaneous link stream [12], recently improved by adapting the Bron–Kerbosch algorithm [6,7].

A **link stream with durations**, or simply link stream, is a triplet $L = (T, V, E)$ where T is a time interval, V a set of nodes and $E \subseteq T \times T \times V \times V$ a set of links such that for all links (b, e, u, v) in E we have $e \geq b$.[1] We call $e - b$ the duration of the link. If there is no distinction between (b, e, u, v) and (b, e, v, u), then L is undirected. If for all (b, e, u, v) in E, $u \neq v$ and for all (b, e, u, v) and (b', e', u, v) in E, $[b, e] \cap [b', e'] = \emptyset$ then L is simple. In the

[1] Though in theory we can consider that time is either discrete or continuous, in all practical cases the datasets have an intrinsic time resolution and time is therefore discrete in practice.

* Corresponding author.
E-mail address: clemence.magnien@lip6.fr (C. Magnien).

https://doi.org/10.1016/j.ipl.2018.01.006

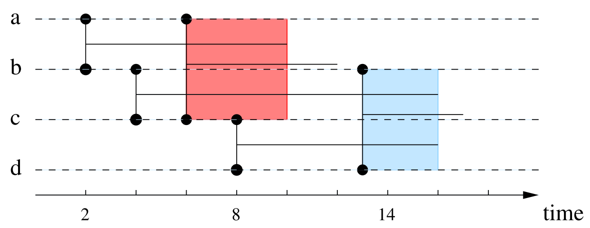

Fig. 1. Maximal cliques in the link stream $L = (T, V, E)$, with $T = [0, 20]$, $V = \{a, b, c, d\}$ and $E = \{(2, 10, a, b), (4, 16, b, c), (6, 12, a, c), (8, 16, c, d), (13, 17, b, d)\}$. Two maximal cliques of L are highlighted: in red, $(\{a, b, c\}, [6, 10])$ is a maximal clique since $[6, 10]$ is the largest interval over which nodes a, b, and c all interact together and they don't interact with d over this time interval. Similarly, $(\{b, c, d\}, [13, 16])$ is a maximal clique, since a does not interact with b, c, d over $[13, 16]$, and there is no larger interval such that b, c, d all interact together. There are no other maximal cliques in L involving 3 nodes, but there are many other maximal cliques in L, such as $(\{c, d\}, [8, 16])$ or $(\{a, c\}, [6, 12])$ for instance.

remainder of this paper, unless explicitly specified, we will consider simple undirected link streams only.

A **clique** in a link stream with durations is a pair $C = (X, [x, y])$ with $X \subseteq V$ and $[x, y] \subseteq T$ such that for all $\{u, v\} \subseteq X$ there is a link (b, e, u, v) in E such that $[x, y] \subseteq [b, e]$. In other words, all pairs of nodes in X are continuously linked together from x to y. A clique $C = (X, [x, y])$ is maximal if there is no other clique $C' = (X', [x', y'])$ such that $C' \neq C$, $X \subseteq X'$ and $[x, y] \subseteq [x', y']$. See Fig. 1 for an illustration.

Cliques in link streams with durations (and Δ-cliques in instantaneous link streams) bring valuable information in the study of different kinds of datasets; for instance they indicate malicious computers coordinating their actions [11]. Likewise, co-presence relations between animals is a key source of insight in ethology [3], and cliques in the link streams with duration modeling such data may indicate significant meetings. Many other fields may benefit from clique computations in link streams with durations in a similar way.

In this paper, we first extend our algorithm for maximal Δ-cliques in instantaneous link streams [12] to enumerate all maximal cliques in link streams with durations. The obtained algorithm is significantly simpler than the previous version, and has a slightly lower complexity; we show that it is possible to use it to enumerate maximal cliques in instantaneous link streams too, making it both more general and more efficient than our previous algorithm. Experiments show that its running time is better than our previous algorithm, as expected, but also that it outperforms the more recent algorithm of Himmel *et al.* [6,7] in several cases of practical interest.

2. Algorithm

Like in [12] our algorithm (Algorithm 1) relies on a set S of previously computed cliques that we call candidates, and a set M of already seen cliques. We initially populate both sets with the trivial clique $(\{u, v\}, [b, b])$, for all links $(b, e, u, v) \in E$ (Line 2) (finding cliques involving only one node is trivial and makes little sense, so we ignore them). Then, our algorithm iteratively picks and processes an element $(X, [x, y])$ from S (Line 4), until S is empty (*while* loop from Line 3 to Line 17). Processing $(X, [x, y])$ consists in searching for nodes $v \notin X$ such that $(X \cup \{v\}, [x, y])$ is a clique (Lines 6 to 10), and for times $y' > y$ such that $(X, [x, y'])$ is a clique (Lines 11 to 15).

For each node v not in X, Line 7 checks that for all u in X, there exists a link (b, e, u, v) in the stream, with $[x, y] \subseteq [b, e]$. If v satisfies this property, then $(X, [x, y])$ is not maximal (Line 8) and if $(X \cup \{v\}, [x, y])$ has not already been seen (Line 9) then we add to both S and M (Line 10).

The value of l computed at Line 11 is the largest time $l > y$ such that $(X, [x, l])$ is a clique. Line 12 checks that this clique is different from $(X, [x, y])$, *i.e.* $l \neq y$. In this case, $(X, [x, y])$ is not maximal (Line 13), and if the new clique $(X, [x, l])$ is new (Line 14) we add it to S and M (Line 15).

If no node v or time l satisfies the conditions above, then the clique $(X, [x, y])$ is maximal and *isMax* is true when we reach Line 16; we add the maximal clique to the output (Line 17).

Algorithm 1 Maximal cliques of a simple undirected link stream with durations.

input: a simple undirected link stream with durations $L = (T, V, E)$
output: the set of all maximal cliques in L involving at least two nodes

```
1: S ← ∅, R ← ∅, M ← ∅
2: for (b, e, u, v) ∈ E: add ({u, v}, [b, b]) to S and to M
3: while S ≠ ∅ do
4:     take and remove (X, [x, y]) from S
5:     set isMax to True
6:     for v in V \ X do
7:         if (X ∪ {v}, [x, y]) is a clique then
8:             set isMax to False
9:             if (X ∪ {v}, [x, y]) not in M then
10:                add (X ∪ {v}, [x, y]) to S and M
11:    l ← min(e : ∃(b, e, u, v) ∈ E such that u, v ∈ X and [x, y] ⊆ [b, e])
12:    if y ≠ l then
13:        set isMax to False
14:        if (X, [x, l]) not in M then
15:            add (X, [x, l]) to S and M
16:    if isMax then
17:        add (X, [x, y]) to R
18: return R
```

Theorem 1 *(Correctness). Given a simple undirected link stream with durations, Algorithm 1 computes the set of all its maximal cliques involving at least two nodes.*

We first show that all the elements in the output of Algorithm 1 are cliques, then that they are maximal, and finally that all maximal cliques are in this output.

Lemma 1. *In Algorithm 1, all elements of S are cliques of L.*

Proof. S initially contains cliques (Line 2) and Line 10 clearly preserves this property. The value of l computed at Line 11 is the smallest value such that there exists a link of the form $(b, l, u, v) \in E$ for any two nodes $u, v \in X$. Since