Algorithmic Challenges in Link Streams: the case of clique computations

Clémence Magnien
work in collaboration with Tiphaine Viard, Matthieu Latapy, Phan Thi Ha Duong, Binh-Minh Bui-Xuan, Pierre Meyer

ComplexNetworks(.fr)
LIP6 (CNRS, Sorbonne Université)

first.last@lip6.fr

July 9th, 2018

Outline

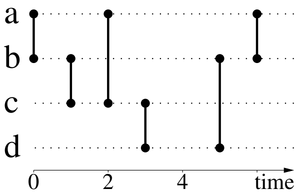
Link streams

Models of **temporal interactions**

$$L = (T, V, E)$$

- $T = [\alpha; \omega]$
- V set of nodes
- $E \subseteq T \times V \otimes V$ set of links

One link = (t, uv)



Two cases of interest

- instantaneous link streams
- link streams with durations

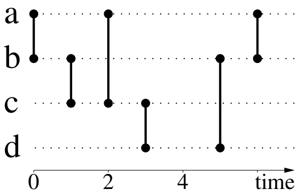
Link streams

Models of **temporal interactions**

$$L = (T, V, E)$$

- $T = [\alpha; \omega]$
- V set of nodes
- $E \subseteq T \times V \otimes V$ set of links

One link $= (t, uv)$



Two cases of interest

- instantaneous link streams
- link streams with durations

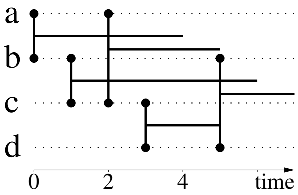
Link streams

Models of **temporal interactions**

$$L = (T, V, E)$$

- $T = [\alpha; \omega]$
- V set of nodes
- $E \subseteq T \times V \otimes V$ set of links

One link $= (b, e, uv)$



Two cases of interest

- instantaneous link streams
- link streams with durations

Definitions

Extensions of graph definitions

- Paths
- (Strongly) Connected components
- Betweenness Centrality
- Cores and shells
- ...

Extensions of algorithms ?

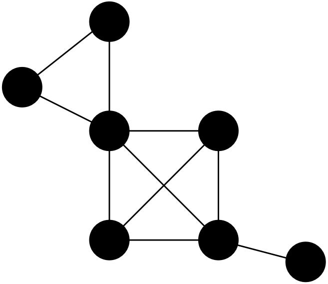
Outline

1. Introduction
2. Maximal cliques in link streams
   - Maximal Δ-cliques in instantaneous link stream
   - Maximal cliques in link streams with durations
3. Link stream edition problems

Clique (in a graph)

$X \subseteq V$

Induced subgraph : all possible links exist

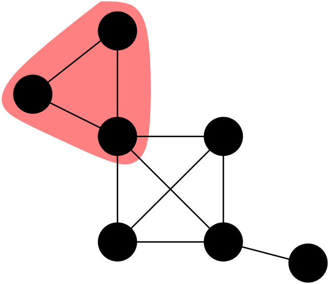


Clique (in a graph)

$X \subseteq V$

Induced subgraph : all possible links exist

Maximal clique : not included in any other clique

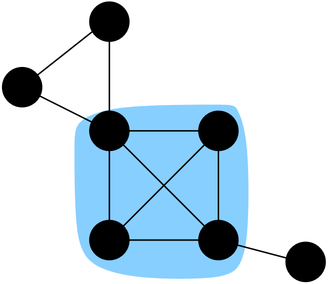


Clique (in a graph)

$X \subseteq V$

Induced subgraph : all possible links exist

Maximal clique : not included in any other clique

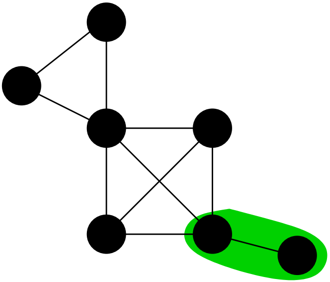


Clique (in a graph)

$X \subseteq V$

Induced subgraph : all possible links exist

Maximal clique : not included in any other clique



Δ-clique in instantaneous link streams

$$(X, [b; e]) \subseteq V \times T$$

Induced sub-stream : all possible links exist all the time

All the time : **at least every** Δ

Maximal if is not included in any other

Examples for $\Delta = 3$:

Signatures of distributed applications, meetings, . . .

Δ-clique in instantaneous link streams

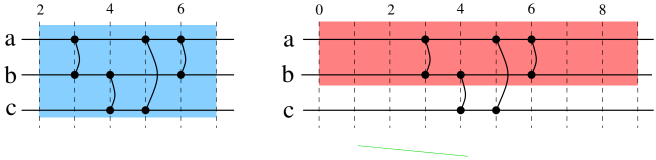
$$(X, [b; e]) \subseteq V \times T$$

Induced sub-stream : all possible links exist all the time

All the time : **at least every** Δ

Maximal if is not included in any other

Examples for $\Delta = 3$:

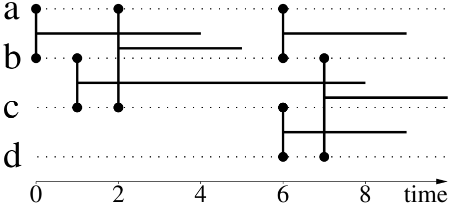


Signatures of distributed applications, meetings, . . .

Cliques in link streams with duration

$$(X, [b; e]), \subseteq V \times T$$

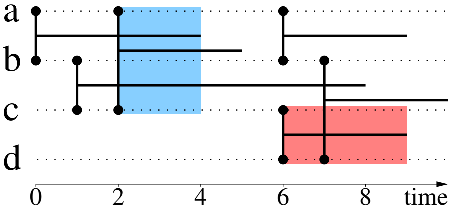
Induced sub-stream : all possible links exist all the time
Maximal if is not included in any other



Cliques in link streams with duration

$$(X, [b; e]), \subseteq V \times T$$

Induced sub-stream : all possible links exist all the time
Maximal if is not included in any other



Outline

Enumerate maximal Δ-cliques in a link stream

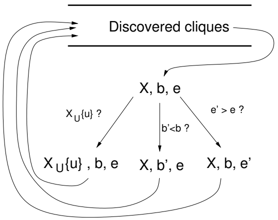
Naive algorithm

Queue Q

- for all $(t, uv) \in E$, $(\{u, v\}, [t, t])$ is a Δ-clique $\longrightarrow Q$

While $Q \neq \emptyset$:

- pop C from Q :
- if a node or time can be added $\longrightarrow Q$
- otherwise C is maximal



Enumerate maximal Δ-cliques in a link stream

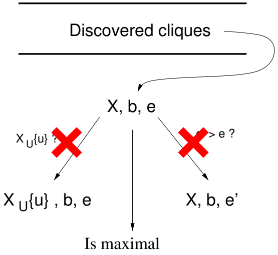
Naive algorithm

Queue Q

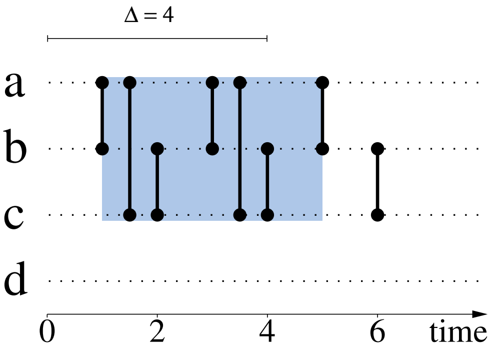
- for all $(t, uv) \in E$, $(\{u, v\}, [t, t])$ is a Δ-clique $\longrightarrow Q$

While $Q \neq \emptyset$:

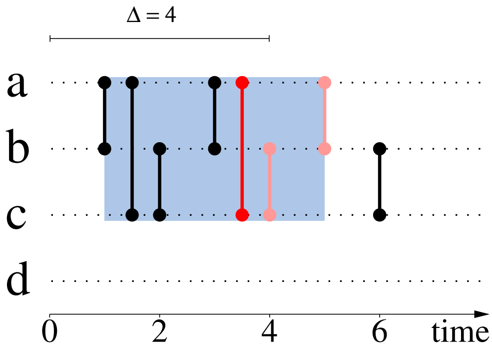
- pop C from Q :
- if a node or time can be added $\longrightarrow Q$
- otherwise C is maximal



Time extension

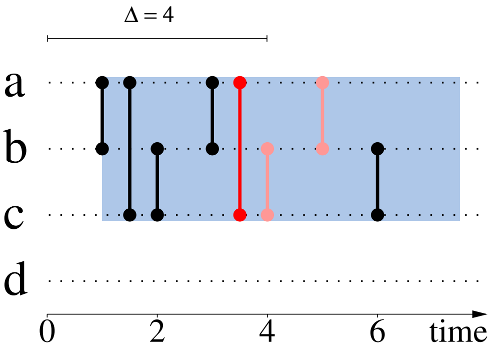


Time extension



- for all links : latest occurrence
- earliest such occurrence

Time extension



- for all links : latest occurrence
- earliest such occurrence
- add Δ

Sketch of proof (1)

1. Initially, all elements of Q are Δ-cliques
2. one step : transforms a Δ-clique into (several) Δ-cliques
3. the output contains only maximal Δ-cliques

Sketch of proof (2)

- All maximal Δ-cliques of L are in the output

Let $C = (X, [b, e])$ be an arbitrary maximal Δ-clique.
(s, uv) : earliest link of C

- $C_0 = (\{u, v\}, [s, s])$
- $C_1 = (\{u, v\}, [s, s + \Delta])$
- ... (add nodes)
- $C_k = (X, [s, s + \Delta])$
- ... (increase time on the right)
- $C_e = (X, [s, e])$
- $C = (X, [b, e])$

Complexity

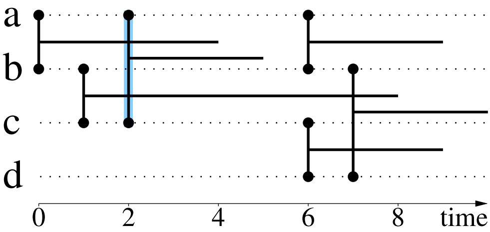
$$\mathcal{O}(2^n n^2 m^3 + 2^n n^3 m^2)$$

Interesting observations

- No relation between n and m
  small n, large m $\longrightarrow$ reasonable running time
- 2^n : All subsets of nodes
  In practice : of nodes linked at the same time
  $\longrightarrow$ Running time increases with Δ

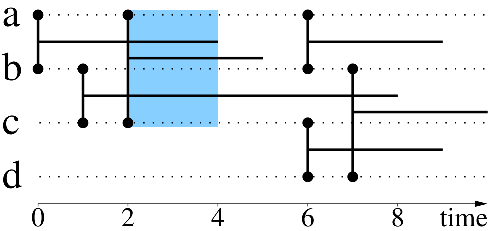
Outline

Same algorithm, except time extension



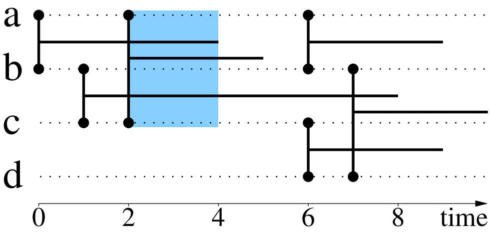
- earliest link end

Same algorithm, except time extension



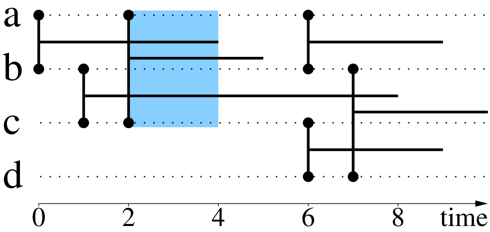
- earliest link end
- Extend time

Same algorithm, except time extension



- earliest link end
- Extend time
- No time extension to the left

Same algorithm, except time extension



- earliest link end
- Extend time
- No time extension to the left

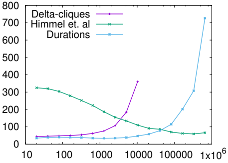
Small complexity gain

from $\mathcal{O}(2^n n^2 m^3 + 2^n n^3 m^2)$ to $\mathcal{O}(2^n n^2 m^2 \log m + 2^n n^3 m^2)$

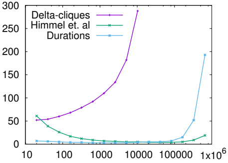
Running times

Algorithms

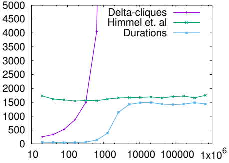
- Δ-cliques in instantaneous linkstreams
- cliques in linkstreams with durations
- Bron-Kerbosch algorithm [HMNS, 2017]



Emails



Highschool



Museum

- Our algorithm fastest for many relevant values of Δ

Case studies

Physical proximity in high school

- Detected structures not observable in the aggregated graph (e.g., students from different classes meeting before class starts)

IP traffic (bipartite)

- Dataset too large to compute all maximal cliques
- Sampling strategy for finding balanced cliques
- Correlation between cliques and malevolent activity

Outline

Link Stream Edition problems

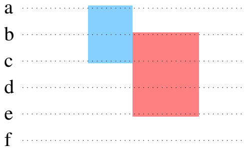
Sparse Split Link Stream Edtion Problem

Given a link stream L and an integer k :

- possible to transform L into a clique (+isolated vertices) in k editions ?

- Well studied in graphs, NP-complete
- Possible to adapt existing algorithms $\longrightarrow$ kernel algorithm

Bi-sparse split linkstream edition



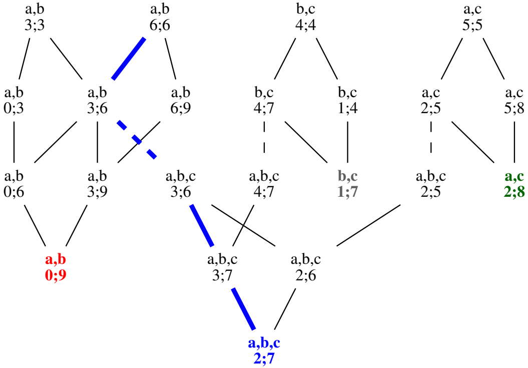
Showed that the problem is fixed parameter tractable and proposed an algorithm

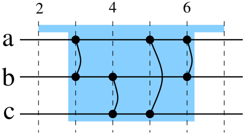
Conclusion and perspectives

- Maximal cliques in link streams
  - Algorithms and code
  - Application to : social interactions description, IP traffic
- Sparse-split and Bi-Sparse-split
  - Algorithms

- Currently
  - clique edge cover problem
  - betwenness centrality
  - strongly connected components
- many graph problems
- Link stream description and understanding
  - anomaly detection

Configuration space





Enumerate maximal cliques in link streams

[Himmel, Molter, Niedermeier, Sorge 2016, 2017]

Adaptation of the Bron-Kerbosch algoritmh to maximal Δ-cliques

Enumerate maximal cliques in graphs

R : clique (not maximal) $P \cup X$: all vertices adjacent to all vertices of R Compute all maximal cliques $\supseteq R$ containing no vertex in X :

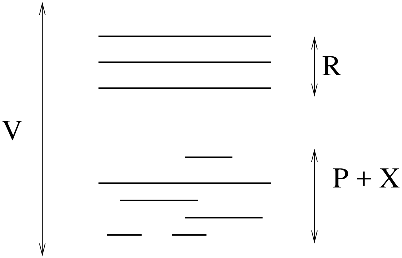
Bron-Kerbosch algorithm

- if $P \cup X = \emptyset \longrightarrow R$ is maximal
- for each $v \in P$
  - `Bron-Kerbosch`$(P \cap N(v), R \cup \{v\}, X \cap N(v))$
  - $P \leftarrow P \backslash \{v\}$
  - $X \leftarrow X \cup \{v\}$

Enumerate maximal cliques in link streams

[Himmel, Molter, Niedermeier, Sorge 2016, 2017]

- (R, I) : time maximal clique
- P, X : sets of (v, I') such that $(R \cup \{v\}, I')$ is a time maximal clique



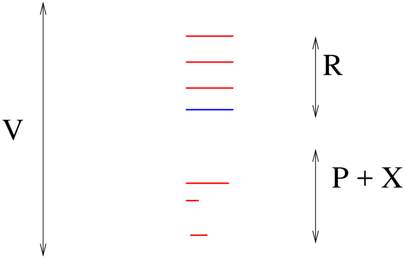
for each $(v, I') \in P$

- add v to R
- Restrict time to I'
- Update P and X

Enumerate maximal cliques in link streams

[Himmel, Molter, Niedermeier, Sorge 2016, 2017]

- (R, I) : time maximal clique
- P, X : sets of (v, I') such that $(R \cup \{v\}, I')$ is a time maximal clique

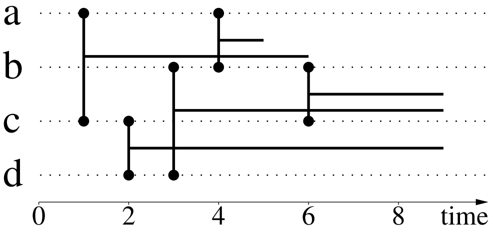


for each $(v, I') \in P$

- add v to R
- Restrict time to I'
- Update P and X

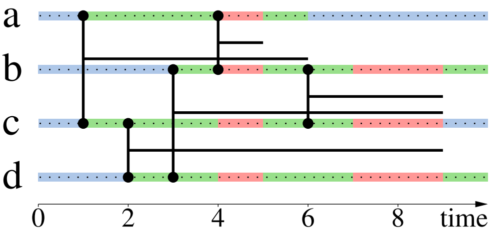
Example : k-core

k-core in a graph : largest induced subgraph s.t. all nodes have degree $\geq t$.



Example : *k*-core

k-core in a graph : largest induced subgraph s.t. all nodes have degree $\geq t$.

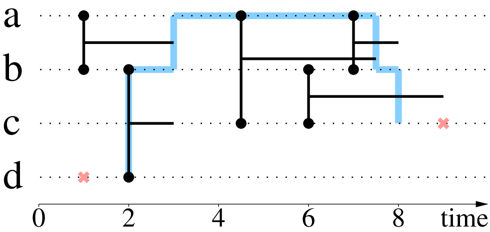


Possible to compute the graph *k*-core at each relevant time-step

Path from (α, u) to (ω, v)

Sequence $(u_0, u_1, t_0), (u_1, u_2, t_1), \dots (u_{-1}, u_k, t_{k-1})$ s.t.

- $u_0 = u, u_k = v$
- $(t_i, u_i, u_{i+1}) \in E$
- $t_i \leq t_{i+1},\ t_0 \geq \alpha, t_{k-1} \leq \omega$



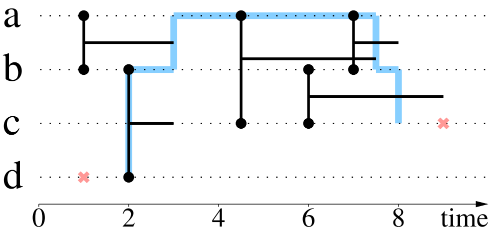
Not possible to consider graphs induced by time instants

Extensions from graph algorithms exist, not direct

Path from (α, u) to (ω, v)

Sequence $(u_0, u_1, t_0), (u_1, u_2, t_1), \dots (u_{-1}, u_k, t_{k-1})$ s.t.

- $u_0 = u, u_k = v$
- $(t_i, u_i, u_{i+1}) \in E$
- $t_i \leq t_{i+1}$, $t_0 \geq \alpha, t_{k-1} \leq \omega$



Not possible to consider graphs induced by time instants

Extensions from graph algorithms exist, not direct