Computing Betweenness Centrality in Link Streams

Clémence Magnien
joint work with Frédéric Simard and Matthieu Latapy

July 2020

Link Streams – definitions

link stream $L = (T, V, E)$
$T = [\alpha, \omega] \subset \mathbb{R}$, V finite set, $E \subseteq T \times V \otimes V$
$(t, uv) \in E \ \Leftrightarrow\ u$ **and** v **are linked at time** t

link segment $[i, j] \times \{uv\} \subseteq E$ with $[i, j]$ maximal
here: finite number of link segments (incl. singletons)

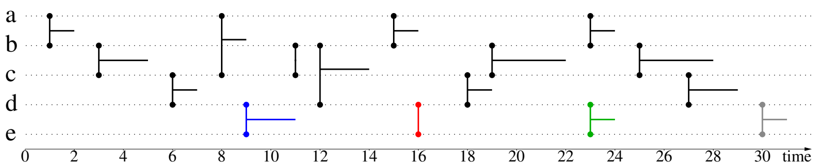
temporal node $(t, u) \in T \times V$

Link Streams – definitions

link stream $L = (T, V, E)$
$T = [\alpha, \omega] \subset \mathbb{R}$, V finite set, $E \subseteq T \times V \otimes V$
$(t, uv) \in E \;\; \Leftrightarrow \;\; u$ **and** v **are linked at time** t

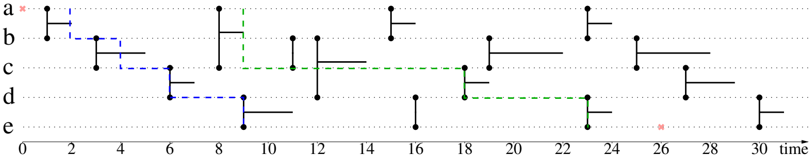
link segment $[i, j] \times \{uv\} \subseteq E$ with $[i, j]$ maximal
here: finite number of link segments (incl. singletons)

temporal node $(t, u) \in T \times V$

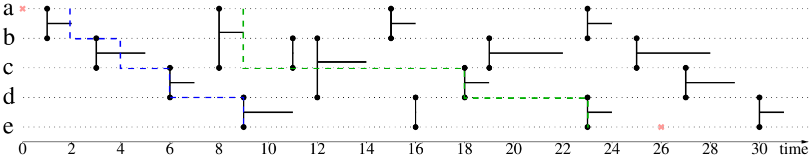


exs: $[9, 11] \times \{de\}$, $\{16\} \times \{de\}$, $[23, 24] \times \{de\}$, $[30, 31] \times \{de\}$

(Shortest Fastest) Paths



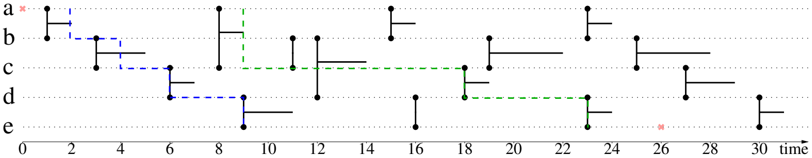
(Shortest Fastest) Paths



path $(x, u) \longrightarrow (y, v)$:
$v_0,\ t_1,\ v_1,\ t_2,\ v_2,\ \dots\ t_k,\ v_k$
such that $u = v_0$, $v_k = v$,
$x \leq t_1 \leq t_2 \leq \cdots \leq t_k \leq y$,
and $(t_i, v_{i-1}v_i) \in E$ for all i

length: k duration: $t_k - t_1$
shortest paths fastest paths
$\hookrightarrow$ **shortest fastest paths (sfp)**

(Shortest Fastest) Paths



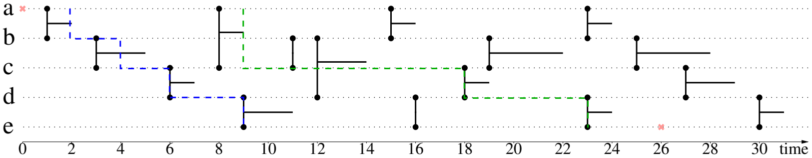
path $(x, u) \longrightarrow (y, v)$:
$v_0,\ t_1,\ v_1,\ t_2,\ v_2,\ \ldots\ t_k,\ v_k$
such that $u = v_0,\ v_k = v$,
$x \leq t_1 \leq t_2 \leq \cdots \leq t_k \leq y$,
and $(t_i, v_{i-1}v_i) \in E$ for all i

length: k — duration: $t_k - t_1$
shortest paths — fastest paths
$\hookrightarrow$ **shortest fastest paths (sfp)**

some paths from $(0, a)$ to $(26, e)$:

$a, 2, b, 4, c, 6, d, 9, e$ **fastest path**, length 4, duration 7, not shortest
$a, 9, c, 18, d, 23, e$ **shortest path**, length 3, duration 14, not fastest
$a, 2, b, 4, c, 6, d, 9, e$ **shortest fastest path (sfp)**

(Shortest Fastest) Paths



path $(x, u) \longrightarrow (y, v)$:
$v_0,\ t_1,\ v_1,\ t_2,\ v_2,\ \ldots\ t_k,\ v_k$
such that $u = v_0$, $v_k = v$,
$x \leq t_1 \leq t_2 \leq \cdots \leq t_k \leq y$,
and $(t_i, v_{i-1}v_i) \in E$ for all i

length: k — duration: $t_k - t_1$
shortest paths — fastest paths
$\hookrightarrow$ **shortest fastest paths (sfp)**

some paths from $(0, a)$ to $(26, e)$:

$a, 2, b, 4, c, 6, d, 9, e$ **fastest path**, length 4, duration 7, not shortest

$a, 9, c, 18, d, 23, e$ **shortest path**, length 3, duration 14, not fastest

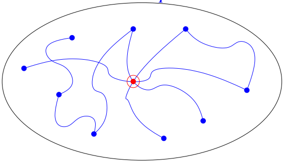
$a, 2, b, 4, c, 6, d, 9, e$ **shortest fastest path (sfp)**

$a, 2, b, 5, c, 6, d, 9, e$ too $\Rightarrow$ **infinity of sfp**

Betweenness Centrality in Link Streams

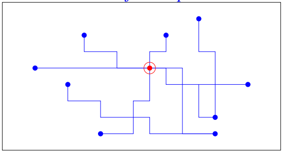
graphs:
vertex,
shortest paths,
all u and v

shortest paths



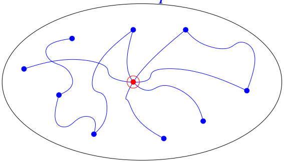
link streams:
temporal vertex,
shortest fastest paths,
all (t, u) and (t', v)

shortest fastest paths

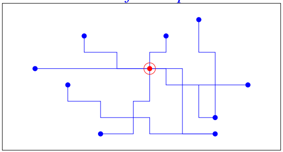


Betweenness Centrality in Link Streams

graphs:
vertex,
shortest paths,
all u and v

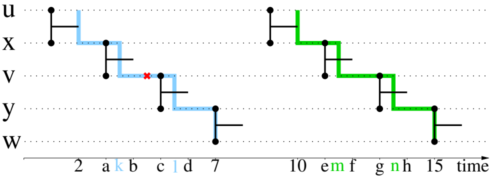


link streams:
temporal vertex,
shortest fastest paths,
all (t, u) and (t', v)



$$\boxed{B(t,v) = \sum_{u \in V, w \in V} \int_{i \in T, j \in T} \underbrace{\frac{\sigma((i,u),(j,w),(t,v))}{\sigma((i,u),(j,w))}}_{\text{fraction of all sfp } (i,u) \longrightarrow (j,w) \text{ that involve } (t,v)} \mathrm{d}i\,\mathrm{d}j}$$

Example



$2 < a \leq b < c \leq d < 7$
and
$10 < e \leq f < g \leq h < 15$

contribution of u and w to $B(t, v)$ with $t \in [b, c]$?

two families of sfp from u to w:

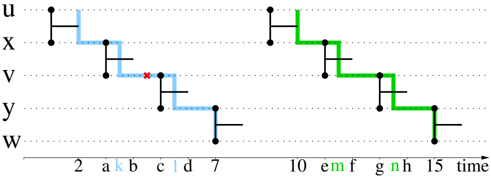
- $u, 2, x, k, v, \ell, y, 7, w$ with $k \in [a, b]$ and $\ell \in [c, d]$

  (**blue family**)
  $(b - a) \cdot (d - c)$ sfp

- $u, 10, x, m, v, n, y, 15, w$ with $m \in [e, f]$ and $n \in [g, h]$

  (**green family**)
  $(f - e) \cdot (h - g)$ sfp

Example



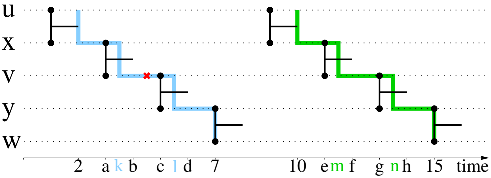
$2 < a \leq b < c \leq d < 7$
and
$10 < e \leq f < g \leq h < 15$

contribution of u and w to $B(t, v)$ with $t \in [b, c]$?

sfp from (i, u) to (j, w):

- blue ones if $i \in [0, 2]$ and $j \in [7, 15[$
- both blue and green ones $i \in [0, 2]$ and $j \in [15, 17]$
- green ones if $i \in]2, 10]$ and $j \in [15, 17]$
- no sfp for all others i and j

Example



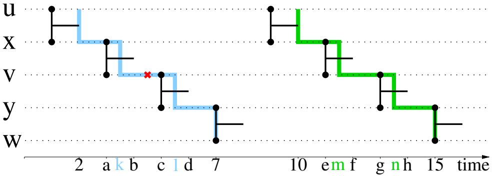
$2 < a \leq b < c \leq d < 7$
and
$10 < e \leq f < g \leq h < 15$

contribution of u and w to $B(t, v)$ with $t \in [b, c]$?

sfp from (i, u) to (j, w):

- blue ones if $i \in [0, 2]$ and $j \in [7, 15[$
  $\hookrightarrow$ . . . . all involve (t, v)
- both blue and green ones $i \in [0, 2]$ and $j \in [15, 17]$
  $\hookrightarrow$ . . . . blue fraction involve (t, v)
  $$\frac{(b-a)\cdot(d-c)}{(b-a)\cdot(d-c)+(f-e)\cdot(h-g)}$$
- green ones if $i \in]2, 10]$ and $j \in [15, 17]$
  $\hookrightarrow$ . . . . none involve (t, v)
- no sfp for all others i and j

Example



$$2 < a \leq b < c \leq d < 7$$
and
$$10 < e \leq f < g \leq h < 15$$

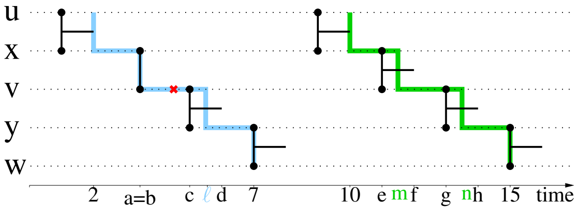
contribution of u and w to $B(t, v)$ with $t \in [b, c]$?

$$\int_0^2 \int_7^{15} 1 \, \mathrm{d}j \, \mathrm{d}i + \int_0^2 \int_{15}^{17} \textit{blue fraction} \; \mathrm{d}j \, \mathrm{d}i$$

$$= 16 + 4 \cdot \textit{blue fraction}$$

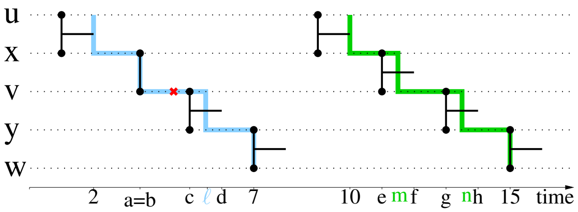
$$\text{blue fraction} = \frac{(b-a)\cdot(d-c)}{(b-a)\cdot(d-c)+(f-e)\cdot(h-g)}$$

Example – what if $a = b$?



how many blue paths? green paths ?

Example – what if $a = b$?



how many blue paths? green paths ?

blue paths:

$u, 2, x, a, v, \ell, y, 7, w$ with $\ell \in [c, d]$

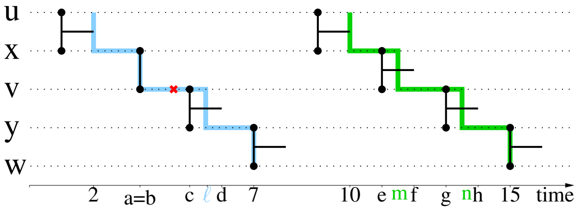
$\hookrightarrow$ volume $(d - c)$, dimension 1

green paths:

$u, 10, x, m, v, n, y, 15, w$ with $m \in [e, f]$ and $n \in [g, h]$

$\hookrightarrow$ volume $(f - e) \cdot (h - g)$, dimension 2

Example – what if $a = b$?



how many blue paths? green paths ?

blue paths:
$u, 2, x, a, v, \ell, y, 7, w$ with $\ell \in [c, d]$
$\hookrightarrow$ volume $(d - c)$, dimension 1

green paths:
$u, 10, x, m, v, n, y, 15, w$ with $m \in [e, f]$ and $n \in [g, h]$
$\hookrightarrow$ volume $(f - e) \cdot (h - g)$, dimension 2

fraction involving (t, v)? 0

except if $e = f$ or $g = h$...

Volumes of Shortest Paths

Identify times of beginning and end of fastest paths
latency pairs from u to w: (s_i, a_i)

Volumes of Shortest Paths

Identify times of beginning and end of fastest paths
latency pairs from u to w: (s_i, a_i)

$\hookrightarrow$ algorithm for volumes of sfp from u to w:

shortest paths within each latency pair

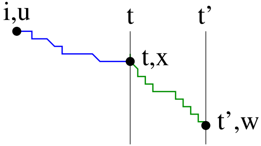
Volumes of Shortest Paths

Identify times of beginning and end of fastest paths
latency pairs from u to w: (s_i, a_i)

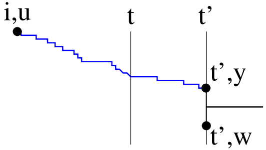
$\hookrightarrow$ algorithm for volumes of sfp from u to w:

shortest paths within each latency pair

using BFS-like from *event time* to event time:



path $(i, u) \longrightarrow (t, x)$
then **path** $(t, x) \longrightarrow (t', w)$



path $(i, u) \longrightarrow (t', y)$
then **jump** $y \rightarrow w$ at t'

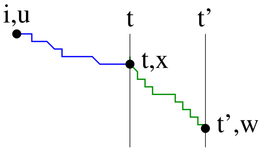
Volumes of Shortest Paths

Identify times of beginning and end of fastest paths
latency pairs from u to w: (s_i, a_i)

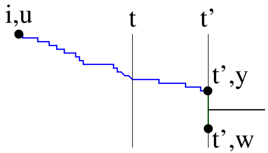
$\hookrightarrow$ algorithm for volumes of sfp from u to w:

shortest paths within each latency pair

using BFS-like from *event time* to event time:



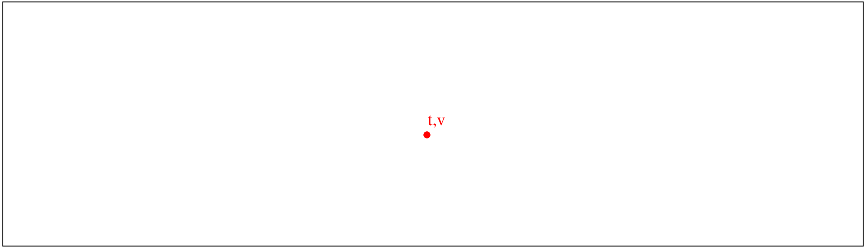
path $(i, u) \longrightarrow (t, x)$
then **path** $(t, x) \longrightarrow (t', w)$



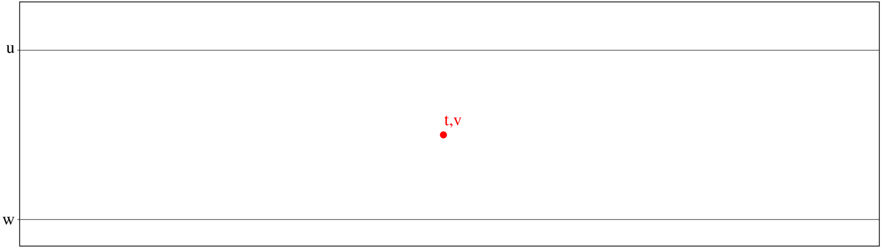
path $(i, u) \longrightarrow (t', y)$
then **jump** $y \rightarrow w$ at t'

+ operation on volumes (considering dimension)

Global Scheme for $B(t, v)$

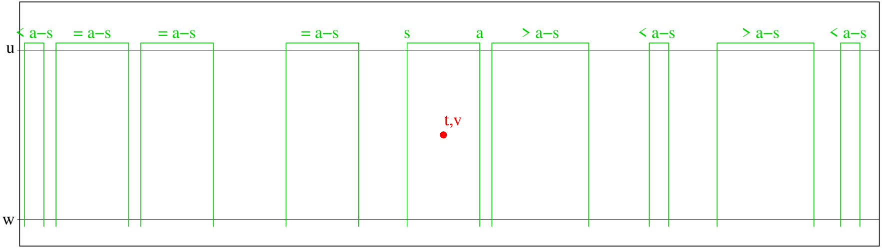


Global Scheme for $B(t, v)$



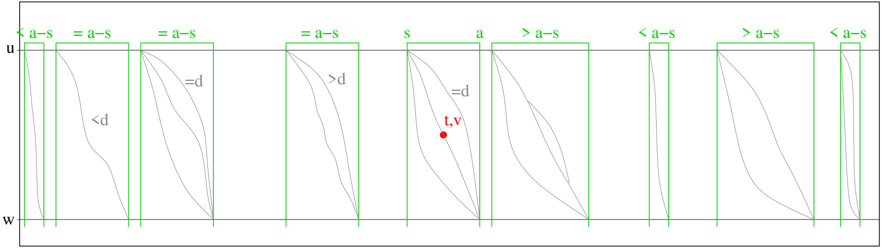
for all u and w:

Global Scheme for $B(t, v)$



for all u and w:
compute **latency pairs for u and w**

Global Scheme for $B(t, v)$

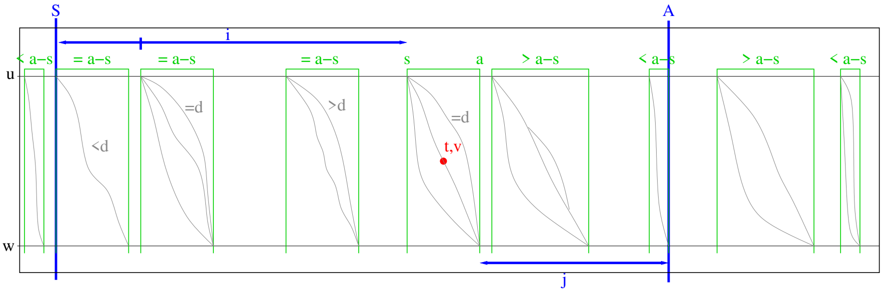


for all u and w:

compute **latency pairs for u and w**

compute length of sp for all latency pairs (sfp)

Global Scheme for $B(t, v)$



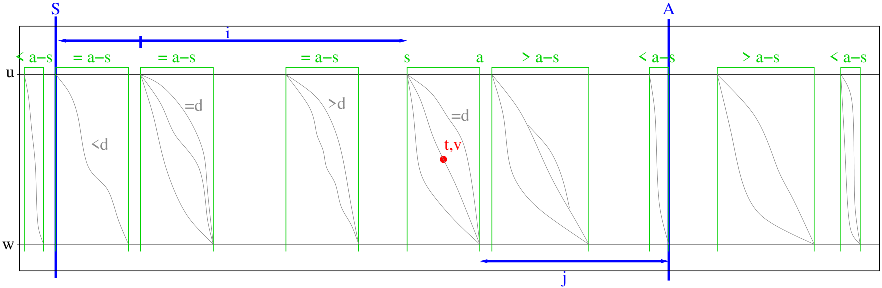
for all u and w:

compute **latency pairs for u and w**

compute length of sp for all latency pairs (sfp)

identify relevant times over which to integrate

Global Scheme for $B(t, v)$



for all u and w:

compute **latency pairs for u and w**

compute length of sp for all latency pairs (sfp)

identify relevant times over which to integrate

integrate over time intervals (sum)

Conclusion and Perspectives

a polynomial algorithm
for
betweenness centrality
in link streams

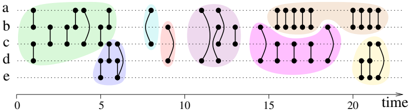
+ implementation

Conclusion and Perspectives

a polynomial algorithm
for
betweenness centrality
in link streams

+ implementation

- improved complexity? easier with discrete time and/or instantaneous links?
- more general cases? stream graphs: dynamic nodes
- betweenness of *all* (t, v)? complexity? approximation?
- community detection



- **postdoc position available**

Notes