Naima *Applied Network Science* (2025) 10:11
https://doi.org/10.1007/s41109-024-00685-5

Applied Network Science

RESEARCH

Open Access

Temporal betweenness centrality on shortest walks variants

Mehdi Naima[1*]

This work is funded in part by the ANR (French National Agency of Research) through the FiT-ANR-19-LCV1-0005 grant.

*Correspondence: mehdi.naima@lip6.fr

[1] LIP6, CNRS, Sorbonne Université, 75005 Paris, France

Abstract

Betweenness centrality has been extensively studied since its introduction in 1977 as a measure of node importance in graphs. This measure has found use in various applications and has been extended to temporal graphs with time-labeled edges. Recent research by Buß et al. and Rymar et al. has shown that it is possible to compute the shortest walks betweenness centrality of all nodes in a temporal graph in $O\left(n^3\,T^2\right)$ and $O\left(n^2\,m\,T^2\right)$ time, respectively, where T is the maximum time, m is the number of temporal edges, and n is the number of nodes. These approaches considered walks that do not take into account contributions from intermediate temporal nodes. In this paper, we study the temporal betweenness centrality on classical walks that we call *passive*, as well as on a variant that we call *active* walks, which takes into account contributions from all temporal nodes. We present an improved analysis of the running time of the classical algorithm for computing betweenness centrality of all nodes, reducing the time complexity to $O\left(n\,m\,T + n^2\,T\right)$. Furthermore, for active walks, we show that the betweenness centrality can be computed in $O\left(n\,m\,T + n^2\,T^2\right)$. We also show that our results hold for different shortest walks variants. Finally, we provide an open-source implementation of our algorithms and conduct experiments on several real-world datasets of cities and contact traces. We compare the results of the two variants on both the node and time dimensions of the temporal graph, and we also compare the temporal betweenness centrality to its static counterpart. Our experiments suggest that for the shortest foremost variant looking only at the first 10% of the temporal interaction is a good approximation for the overall top ranked nodes.

Keywords: Graph algorithms, Experimental algorithms, Betweenness centrality, Temporal graphs, Shortest paths, Time centrality, Restless walks

Introduction

Betweenness centrality is a well-known centrality measure in static graphs that aims to identify central nodes in a graph. Centrality measures assign a value to each node (or edge) based on their importance (centrality). In a static graph, the betweenness centrality of a node is based on the number of shortest paths passing through that node. It was introduced by Freeman in Freeman (1977). This centrality has been studied extensively in the literature and is a classical measure in network analysis used in a variety of domains such as social networks Burt (2004), transports Puzis et al. (2013), biology Narayanan (2005); Yoon et al. (2006) and scientific collaboration networks Leydesdorff

(2007). Additionally, betweenness centrality has been utilized as an efficient method for graph partitioning and community detection Girvan and Newman (2002). Brandes in Brandes (2001) introduced a method for computing betweenness centrality of a whole graph in $O(n\,m + n^2)$ which remains the fastest known algorithm.

Recently, betweenness centrality has been extended to dynamic graph formalisms such as temporal graphs Kostakos (2009) and stream graphs Latapy et al. (2018). The generalization of betweenness centrality to a temporal setting is not unique, and many optimality criteria have been considered in the literature Buß et al. (2020); Rymar et al. (2021); Tsalouchidou et al. (2020); Tang et al. (2010); Kim and Anderson (2012); Latapy et al. (2018), including shortest walks, fastest walks, foremost walks, and shortest fastest walks. However, for this paper, we only focus on the shortest walks (minimal number of hops) criteria which has also been studied in Buß et al. (2020); Rymar et al. (2021) as it is the most straightforward generalization of the static case. It is then possible to define the betweenness centrality of a node v at time t by:

$$B(v,t) = \sum_{s \neq v \neq z \in V} \frac{\sigma_{sz}(v,t)}{\sigma_{sz}},$$

where $\frac{\sigma_{sz}(v,t)}{\sigma_{sz}}$ is the fraction of shortest temporal walks from s to z that pass through node v at time t. Recent results on temporal betweenness centrality, tried with success to adapt Brandes algorithm to the temporal setting Buß et al. (2020); Rymar et al. (2021). For shortest walks their approach lead to time complexities of $O\big(n^3\,T^2\big)$ and $O\big(n^2\,m\,T^2\big)$ to compute the betweenness of a whole temporal graph. However, their algorithms considered only what we call *passive* temporal walks in which the walk only exists when it arrives at a certain temporal node, and moreover, they did not apply Brandes algorithm to its full extent as we shall see.

In a temporal walk, when there is a delay between steps like starting from a node u, the walk transitions to node u_1 at time t_1, then later transitions to node u_2 at time t_2. Many existing works consider that such a walk contributes to the betweenness of u_1 only at time t_1, while we investigate the more general and more natural version in which the walk contributes to the betweenness of u_1 for all times between t_1 and t_2. Indeed, removing u_1 at any of these times makes the walk unfeasible. To this end, we consider both what we call *passive* and *active* shortest walks, so that active walks exist all along a node until leaving it while passive walks correspond to the more classical version. For the classical passive shortest walks, we improve the time analysis of Buß et al. (2020) and show that the Betweenness centrality of the whole graph can be computed in $O\big(n\,m\,T + n^2\,T\big)$. This bound increases to $O\big(n\,m\,T + n^2\,T^2\big)$ if considering active shortest walks. We also show that these bounds are still true for shortest k -restless walks where it is not possible to stay more than k time units on the same node and for shortest foremost walks where we want to reach a node as soon as possible. For all stated criteria the results also hold on their *strict* versions where traversing a node takes one time unit. Our time analysis results show that we can use Brandes approach to its full extent in the temporal setting since when the temporal graph is static (i.e its edges exist at only one timestamp) our analysis reduces to the state of art algorithm on static graphs Brandes (2001). In fact active walks were considered in Latapy et al. (2018) and Tang et al. (2010) but not on shortest walks (number of transitions) and we also seek to have a *systematic* study of all these shortest walks variants as was the case in Buß et al. (2020); Rymar et al. (2021) by designing a single algorithm for all these variants which was not the aim of these works.

We also provide an open-source implementation in C++ and use it to assess the differences between active and passive variants on real-world temporal graphs in both their *node* and *time* dimensions. On the *node dimension* of the temporal graph we compare temporal betweenness centrality to the static betweenness centrality computed on the aggregated graph. Our experiments show that the temporal and static betweenness centrality rankings of nodes are close to each others with the static betweenness running 100 times faster. On the *time dimension* our experiments show that the active variant that we propose gives more importance to central times in contrast with the passive classical variant where the first times of the graph are the most important. Finally, our experiments suggest that for the shortest foremost variant looking only at the first 10% of the temporal interactions is a good approximation for the overall top ranked nodes.

The paper is organized as follows, in Sect. 2 we introduce our formalism that is a modified version of Rymar et al. (2021). We start by defining active and passive walks and giving a motivation for the study of active walks. We end this section by defining the betweenness centrality of a temporal node and give the statement of our main Theorem 1 followed by a discussion of the results. After that in Sect. 3 we give the main ideas and algorithms to prove our results with more details given in Sect. 5. Finally, Sect. 4 presents our experimental results. We mainly focus on the differences of behaviours between active and passive walks and show that the rankings of temporal nodes are moderately correlated on real-world datasets. We end this paper with some perspectives in Sect. 6.

Formalism

We use a formalism close to the ones used in Buß et al. (2020); Rymar et al. (2021). We define a directed temporal graph G as a triple $G = (V, \mathcal{E}, T)$ such that V is the set of vertices, $T \in \mathbb{N}$, is the maximal time step with $[T] := \{1, \ldots, T\}$ and $\mathcal{E} \subseteq V \times V \times [T]$ is the set of temporal arcs (transitions). We denote by $n := |V|$ and $m := |\mathcal{E}|$. We call $V \times [T]$ the set of temporal nodes. Then $(v, w, t) \in \mathcal{E}$ represents a temporal arc from v to w at time t.

Definition 1 *(Temporal walk)* Given a temporal graph $G = (V, \mathcal{E}, T)$, a temporal walk W is a sequence of transitions $e \in \mathcal{E}^k$ with $k \in \mathbb{N}$, where $e = (e_1, \ldots, e_k)$, with $e_i = (u_i, v_i, t_i)$ such that for each $1 \leq i \leq k-1$ of $u_{i+1} = v_i$ and $t_i \leq t_{i+1}$.

The length of a temporal walk W denoted $len(W)$ is its number of transitions. We also denote by $arr(W)$ the time of the last transition of W. We can associate a type of walks to consider on a temporal graph. We will study in this paper two types of walks on temporal graphs that are called active and passive walks. We will denote the type of walks considered on a temporal graph G by

$$\mathrm{type}(G) \in \{\mathrm{act}, \mathrm{pas}\},$$

where *pas* stands for passive and *act* for active. The important difference between active and passive walks is that, a passive walk only exists on node transitions, therefore a passive walk that arrive to v at time t and leaves v at t' only exists on node v for a single time t, while an active walk exists on v for all times $t \leq i \leq t'$. This difference is formally defined in the following definition.

Definition 2 *(Visited temporal nodes)* For a temporal graph G, fix a walk type. Let W be a temporal walk such that $len(W) = k$ and let $k > 0$. Then the list of visited temporal nodes $\mathcal{V}(W)$ is given by:

$$\mathcal{V}(W) = \begin{cases} [(u_1, t_1)] + [(v_i, t_i) \mid 1 \leq i \leq k], & \text{if type(G)} = \text{pas} \\ [(u_1, t_1)] + \left(\biguplus_{i=1}^{k-1} [(v_i, t) \mid t_i \leq t \leq t_{i+1}] \right) + [(v_k, t_k)] & \text{otherwise} \end{cases}$$

where $+$ denotes list concatenation and $\biguplus$ is used for concatenation of several lists.

We will denote by $W[-1]$ the last visited temporal node corresponding to the last element in $\mathcal{V}(W)$. We can denote a temporal walk using an arrow notation. For instance $W = [(a, b, 1), (b, c, 5)]$ of the temporal graph in Fig. 1 by $W = a \xrightarrow{1} b \xrightarrow{5} c$. See Fig. 1 for an example of these concepts. This distinction between active and passive walks is important since the authors of recent results in this line of research Buß et al. (2020); Rymar et al. (2021) consider only passive walks. A temporal walk is called a path if each node $v \in V$ in the list of visited nodes appears exactly once. Moreover, a temporal walk W is a strict temporal walk if for each transition time label is strictly larger than the previous one, that is for $2 \leq i \leq k, t_i > t_{i-1}$. Otherwise, the temporal walk is a non-strict temporal walk. Finally, for $k \in \mathbb{N}$, a temporal walk is k-restless if the difference between two consecutive transitions time stamps $t_i - t_{i-1} \leq k$.

Example 1 *(Motivation example for the study of active walks)* Consider a message passing temporal network $G = (V, \mathcal{E}, T)$ where V is a set of machines (computers or routers) and a temporal arc $(u, v, t) \in \mathcal{E}$ corresponds to a message sent from u to v at time t. Suppose that Fig. 1 represents this graph. Take for instance node c at time 4. At this time node c is retaining a message that arrived from b at time 2 and another one that arrived from d at time 3. Therefore, node c at time 4 is retaining important information. It is important to make sure that machine c is not disconnected from the network at this time to carry this information to other nodes. However, considering the classical passive of temporal walks in Burt (2004); Rymar et al. (2021) $(c, 4)$ is not visited by any temporal

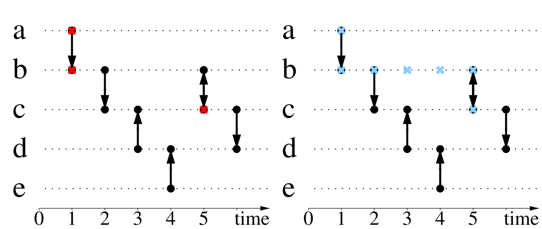

Fig. 1 A temporal graph having nodes $V = \{a, b, c, d\}$ and $T = 7$ with arrows representing the set $\mathcal{E}$. The walk $W = [(a, b, 1), (b, c, 5)]$ can be denoted as $W = a \xrightarrow{1} b \xrightarrow{5} c$. blue

walk in the graph and get 0 value for its betweenness centrality. On the other hand this information is entirely captured by active variant that we propose.

A walk W is an $s-v$ walk if W starts in node s and ends in node v, we denote by W_{sv} the set of all $s-v$ walks. In this paper we consider 3 variants of shortest temporal walks. Shortest here refer to minimizing the number of transitions (length) of the temporal walk. These variants are:

- Shortest walks (sh) which minimize the walk length over all walks going from a node to another,
- Shortest k-restless walks(sh-k) which minimize the walk length over all k-restless walks going from a node to another and
- Shortest foremost (fm) walks which minimize the walk length over all walks going from one node and arriving the earliest in time to the other.

For each one of these criteria, we need to define the shortest possible length of temporal walks from a node to another.

$$c_s^{sh}(v) = \min_{W \in W_{sv}} (len(W)) \quad \text{(shortest)} \tag{1}$$

$$c_s^{sh-k}(v) = \min_{\substack{W \in W_{sv}, \\ W \text{ is } k-restless}} (len(W)) \quad \text{(shortest } k\text{-restless)} \tag{2}$$

$$c_s^{fm}(v) = \min_{W \in W_{sv}} (arr(W) \cdot n + len(W)) \quad \text{(shortest foremost)} \tag{3}$$

Note that the definition of $c_s^{fm}(v)$ ensures only considering walks arriving first and then minimizing over their lengths. We will denote by $c^\star(W)$ the cost of a temporal walk where $\star \in \{sh, sh-k, fm\}$. Then $c^{sh}(W) = len(W)$, $c^{sh-k}(W) = len(W)$ if W is $k-restless$ and ∞ otherwise and $c^{fm}(W) = (arr(W) \cdot n + len(W))$.

Remark 1 Shortest walks are necessarily paths while this is not true in general for shortest k-restless walks. In fact, finding a k-restless path has been shown to be NP-hard in Casteigts et al. (2021). As a consequence we will use the term walks in general because we want to encompass all variants.

For an active temporal walk W we will denote by W_t with $t > arr(W)$ the extension of W on its last node to t. Formally, $\mathcal{V}(W_t) = \mathcal{V}(W) + [(v_k, t') \mid arr(W) < t' \leq t]$ where v_k is the arrival node of W. For example, on the graph of Fig. 1, $W = a \xrightarrow{1} b \xrightarrow{5} c$. Then $\mathcal{V}(W_7) = [(a,1),(b,1),(b,2),(b,3),(b,4),(b,5),(c,5),(c,6),(c,7)]$. Now we can define:

Definition 3 *(Set of shortest walks)* Let $G = (V, \mathcal{E}, T)$ be a temporal graph and fix a cost. Then

$$\mathcal{W}^\star = \begin{cases} \bigcup_{s,z\in V, s\neq z}\{W | W \in W_{sz}, c(W) = c_s^\star(z))\}, & \text{if } type(G) = pas \\ \bigcup_{s,z\in V, s\neq z}\{W_T | W \in W_{sz}, c(W) = c_s^\star(z))\}, & \text{otherwise.} \end{cases}$$

where $\star \in \{sh, sh-k, fm\}$.

The reason for the extension of the walks to the last time will be made clear in Sect. 5.

Remark 2 If we allow $k = \infty$ in the k-restless setting, then $c_s^{sh}(v) = c_s^{sh-\infty}(v)$ and $\mathcal{W}^{sh} = \mathcal{W}^{sh-\infty}$ since the shortest walk criteria allows all walks regardless of difference in transition time between edges. Therefore, we will only focus on showing our results on k-restless criteria for $k \in \mathbb{N} \cup \{\infty\}$.

We see that $\mathcal{W}$ is the set of shortest walks between any pair of nodes, it keeps only walks with an overall shortest value.

Definition 4 Let $G = (V, \mathcal{E}, T)$ be a temporal graph. Fix a walk type, a cost and let $s, v, z \in V$ and $t \in [T]$. Let $\mathcal{W}^\star$ be as in Definition 3. Then,

- σ_{sz} is the number of $s - z$ walks in $\mathcal{W}^\star$.
- $\sigma_{sz}(v, t)$ is the number of $s - z$ walks $W \in \mathcal{W}^\star$, such that W passes through (v, t) that is $(v, t) \in \mathcal{V}(W)$ according to Definition 2.

We note that σ_{sz} depends only on the cost considered while $\sigma_{sz}(v, t)$ depends on both the cost and the walk type considered.

Definition 5 Given a temporal graph $G = (V, \mathcal{E}, T)$, a walk type and a cost. We define

$$\delta_{sz}(v,t) = \begin{cases} 0 & \text{if } \sigma_{sz} = 0, \\ \dfrac{\sigma_{sz}(v,t)}{\sigma_{sz}} & \text{otherwise.} \end{cases} \quad \delta_{s\cdot}(v,t) = \sum_{z\in V} \delta_{sz}(v,t).$$

Definition 6 *(Betweenness centrality of a temporal node)* Given a temporal graph $G = (V, \mathcal{E}, T)$ and a walk type. The betweenness centrality of node v at time t is:

$$B(v,t) = \sum_{\substack{s,v,z \in V, \\ s \neq v \neq z}} \delta_{sz}(v,t).$$

According to our definitions there are 2 walk types and 3 costs considered. Therefore, we have 6 different variants that can be considered corresponding to any combination of walk type and cost. Now, from the preceding we define

$$\hat{B}(v,t) = \sum_{s,z\in V} \delta_{sz}(v,t) \implies \quad \hat{B}(v,t) = \sum_{s\in V} \delta_{s\cdot}(v,t).$$

The quantities $B(v, t)$ and $\hat{B}(v,t)$ are related through:

$$B(v,t) = \hat{B}(v,t) - \sum_{w \in V} (\delta_{vw}(v,t) + \delta_{wv}(v,t)) = \hat{B}(v,t) - \delta_{v\cdot}(v,t) - \sum_{w \in V} \delta_{wv}(v,t) \quad (4)$$

For instance on Fig. 1, considering passive shortest walks (approach used in Buß et al. (2020); Rymar et al. (2021)), $B(b,1) = 2$, and $\forall t > 1, B(b,t) = 0$ while if we consider active shortest walks we have $B(b,1) = B(b,2) = 2, B(b,3) = B(b,4) = 1$ showing that the active version takes into account contributions from intermediate temporal nodes in shortest paths while it is not true for the passive version.

From the betwenness centrality of a temporal node we can get an overall betweenness centrality of a node v and an overall betweenness centrality of a time.

Definition 7 *(Overall betweenness of a node and overall betweenness of a time)*

$$B(v) = \sum_{t \in [T]} B(v,t), \quad B(t) = \sum_{v \in V} B(v,t).$$

Results

While the authors of Buß et al. (2020); Rymar et al. (2021) focus on computing $B(v)$ for all $v \in V$, we focus on the computation of $B(v, t)$. Our main result is the following:

Theorem 1 *Let $G = (V, \mathcal{E}, T)$ be a temporal graph. For passive walks, the betweenness centrality of all temporal nodes can be computed in $O(n\,m\,T + n^2\,T)$ considering shortest, shortest k-restless and shortest foremost walks. For active walks, the betweenness centrality of all temporal nodes can be computed in $O(n\,m\,T + n^2\,T^2)$ considering shortest walks. Both results hold for strict and non-strict versions.*

Proof We leave the proof of Theorem 1 to the end of Sect. 3. □

Discussion The authors of Buß et al. (2020); Rymar et al. (2021) showed that for passive walks, the overall betweenness of nodes $B(v)$ (not temporal nodes) can be computed in $O(n^3\,T^2)$ and $O(n^2\,m\,T^2)$ respectively. Since the maximal number of temporal arcs is $(n-1)^2 T$, our bounds are always better than the previously known ones. Additionally, in the introduction we mentioned using *Brandes approach to its full extent*, since these previous approaches when $T = 1$ reduce to $O(n^3)$ and $O(n^2\,m)$ while our analysis lead to $O(nm + n^2)$. Therefore our approach leads to the static optimal time algorithm if the temporal graph is static. Table 1 summarizes our results compared to the other two when taking the overall betweenness of nodes.

For active versions, only shortest walks can be computed with our algorithm. While shortest foremost and shortest k-restless (for $k < \infty$) can not be computed using our algorithm. In Sect. 6 and Remark 3 we discuss why the results do not hold.

Main algorithms and proofs

According to Remark 2 we only need to consider 3 variants in our proofs that are (active, ∞-restless), (passive, k-restless) and (passive, shortest foremost) since $k = \infty$ covers the classical shortest walks criteria. We will prove our results for both (active,

Table 1 Improvement of previously known results by Theorem 1

	Rymar et al. (2021)	Buß et al. (2020)	Theorem 1
Shortest (passive)	$O(n^2 mT^2)$	$O(n^3 T^2)$	$O(nmT + n^2T)$
Shortest k-restless (passive)	$O(n^2 mT^2)$	–	$O(nmT + n^2T)$
Shortest foremost (passive)	$O(n^2 mT^2)$	$O(n^3 T^2)$	$O(nmT + n^2T)$
Shortest (active)	–	–	$O(nmT + n^2T^2)$

The results on active walks were not studied in this form to our knowledge. All results hold for non-strict and strict variants

∞-restless), (passive, k-restless) walks and in Sect. 5.2 give the necessary details for (passive, shortest foremost) variant.

Optimal passive walks

We denote by $W^{k,pas}_{s(v,t)}$ the set of passive k-restless $s-(v,t)$ walks which is:

$$\begin{aligned} W^{k,pas}_{s(v,t)} =& \{W \mid W \in W_{sv}, arr(W) = t, W \text{ is} k\text{-restless } \} \\ & \cup \{\epsilon \mid s = v, (s,w,t) \in \mathcal{E} \text{ for some } w \in V\}, \end{aligned}$$

where ϵ denotes the empty walk.

Let $G = (V, \mathcal{E}, T)$ be a temporal graph. Fix a source $s \in V$ and a walk type. Then, for every temporal node $(v,t) \in V \times [T]$ we define the optimal cost from s to temporal node (v, t). Formally:

$$c^{k,pas}_s(v,t) = \min_{W \in W^{k,pas}_{s(v,t)}} \left(c^{sh-k}(W)\right). \tag{5}$$

If the set of $s-v$ walks is empty then $c^{k,pas}_s(v,t) = \infty$. Now, the overall optimal values from s to any time on node v as defined in Eq. (2) can be computed as:

$$c^{sh-k}_s(v) = \min_{t \in [T]} \left(c^{k,pas}_s(v,t)\right). \tag{6}$$

Finally, for fixed walk type we say that a temporal $s-(v,t)$ walk W is an optimal $s-(v,t)$ k-restless walk if $c^{sh-k}(W) = c^{k,pas}_s(v,t)$. Similarly an $s-z$ walk W is an optimal $s-z$ k-restless walk if $c^{sh-k}(W) = c^{sh-k}_s(z)$.

We denote by $\mathcal{W}^{k,pas}_{s(v,t)}$ the set of optimal $s-(v,t)$ walks. That is

$$\mathcal{W}^{k,pas}_{s(v,t)} = \{W \mid W \in W^{k,pas}_{s(v,t)}, c^{sh-k}(W) = c^{k,pas}_s(v,t).\} \tag{7}$$

The predecessor graph in temporal settings has been used in Rymar et al. (2021); Buß et al. (2020). Here, we extend its definition to encompass active walks as well.

The two major steps of the proof are the following. First step is to build a predecessor graph from a fixed node $s \in V$ efficiently. This predecessor graph allows then to compute the contributions of node $s \in V$ to the betweenness centrality of all other nodes. Second step is to find a recurrence that allows to compute the aforementioned contributions efficiently. Propositions 9 and 3 correspond to these steps.

Definition 8 *(predecessor set, successor set)* Let $G = (V, \mathcal{E}, T)$ be a temporal graph fix a source $s \in V$, then for all $w \in V, w \neq s$, let $\mathcal{W}^{k,pas}_{s,(v,t)}$ be the set of optimal $s-(v,t)$ walks:

$$\begin{aligned} pre_s(w,t') =& \{(v,t) \in V \times [T] \mid \exists\, m \in \mathcal{W}^{k,pas}_{s(w,t')}, m = s \xrightarrow{t_1} \ldots \xrightarrow{t} v \xrightarrow{t'} w\} \\ & \cup \{(s,t') \mid \exists\, m \in \mathcal{W}^{k,pas}_{s(w,t')}, m = s \xrightarrow{t'} w\}. \end{aligned}$$

The successor set of a node $succ_s(w,t') = \{(v,t) \mid (w,t') \in pre_s(v,t)\}$.

A consequence of our framework is that the empty walk is an exact $s-(s,t)$ whenever there exists at least one node $w \in V$ with $(s,w,t) \in \mathcal{E}$. Then $c_s(s,t) = 0$.

Definition 9 *(Predecessor graph)* The predecessor graph $G_s = (V_s, E_s)$ is the directed graph obtained from pre_s, whose arcs are given by

$$E_s = \{((v,t),(w,t')) \mid (v,t) \in pre_s(w,t')\},$$

and its vertices V_s are the ones induced by E_s.

An example of the predecessor graph for passive shortest walks (i.e $k = \infty$) is depicted on Fig. 2. As we shall see, a path in the predecessor graph represents a unique walk in the temporal graph. The next Proposition gives the relationship between these quantities.

Lemma 1 *Fix $s \in V$. There is a one-to-one correspondence between a path p in the predecessor graph G_s starting from (s, t) for some $t \in [T]$ and ending in a temporal node (v,t') and an optimal $s-(v,t')$ walk.*

Lemma 2 *The predecessor graph G_s from $s \in V$ is acyclic.*

We define $\sigma_{s,(v,t)} = |\mathcal{W}^{k,pas}_{s,(v,t)}|$ which corresponds to the number of optimal $s-(v,t)$ walks.

Proposition 2 *For any temporal node (v, t), there holds that:*

$$\sigma_{s(v,t)} = \begin{cases} 0 & \text{if } (v,t) \neq V_s, \\ 1 & \text{if } (v,t) \in V_s \text{ and } v = s, \\ \sum\limits_{(w,t') \in pre_s(v,t)} \sigma_{s(w,t')} & \text{otherwise.} \end{cases} \tag{8}$$

Finally it is straight forward to compute for all $v \in V$ and $t \in [T]$, σ_{sv} and $\sigma_{sv}(v,t)$ from the preceding:

$$\sigma_{sv} = \sum_{\substack{t \in [T] \\ c^{k,pas}_s(v,t) = c^{sh-k}_s(v)}} \sigma_{s(v,t)}, \quad \sigma_{sv}(v,t) = \begin{cases} \sigma_{s(v,t)} & \text{if } c^{k,pas}_s(v,t) = c^{sh-k}_s(v) \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

The main result allowing to efficiently compute the contributions from a node s to the betweenness centrality of all others is a an extension of the recurrence found by

Brandes in Brandes (2001). This recurrence has been adapted to temporal graphs in Buß et al. (2020); Rymar et al. (2021), in this section we give the recurrence for the passive case and in Sect. 3.2 we extend it further for active walks. Let $G_s = (V_s, E_s)$ be the predecessor graph of node s, then:

Proposition 3 (General passive contribution) *Fix a node $s \in V$, considering passive walks then for $(v,t) \in V_s$*

$$\delta_{s\cdot}(v,t) = \delta_{sv}(v,t) + \sum_{\substack{(w,t') \in succ_s(v,t) \\ t' \geq t}} \frac{\sigma_{s(v,t)}}{\sigma_{s(w,t')}} \delta_{s\cdot}(w,t'). \tag{10}$$

Algorithm 1 Predecessor graph from node s

```
Input: G = (V, E, T) : a temporal graph, s : a node in G, k : maximal waiting time for k-restless walks.
Output: A dictionary dist containing shortest values c_s(v, t) to temporal nodes. A dictionary pre that
    contains the set of predecessor temporal nodes.
function Temporal_BFS(G,s,k)
    pre, dist, Q = initalization(G, s)
    Q' = Empty_Queue(), ℓ = 1
    while Q ≠ ∅ do
        for (a, t) in Q do
            for a -t'-> b ∈ E such that (not ((s = a) and (t' ≠ t))) do
                if (s = a) or ((t' ≥ t) and (t' − t) ≤ k) then
                    relax(a, b, t, t', pre, dist, Q', ℓ, k)
                end if
            end for
        end for
        ℓ = ℓ + 1
        (Q, Q') = (Q', Empty_Queue())
    end while
    return pre, dist
end function
function initialization(G,s)
    Q = Empty_Queue()
    dist[v] = {t : ∞, ∀t ∈ {0, ..., T}} ∀v ∈ V
    pre[v] = {t : {}, ∀t ∈ {0, ..., T}} ∀v ∈ V
    for t ∈ {t' | ∃w ∈ V, s -t'-> w ∈ E} do
        dist[s][t] = 0, pre[s][t] = {(nil, nil)}
        Q.enqueue((s, t))
    end for
    return pre, dist, Q
end function
function relax(a, b, t, t', dist, pre, Q, ℓ, k)
    if (dist[b][t'] = ∞) or (dist[b][t'] ≥ ℓ and |pre[b][tp]| = 0) then
        dist[b][t'] = ℓ, pre[b][t'] = {}
        Q.enqueue((b, t'))
        for t'' ∈ {r | ∃w, w -r-> b ∈ E, (r > t' and (r − t') ≤ k)} do   ▷ for passive walks, ignore this loop.
            relax_extend(b, t'', pre, dist, ℓ)
        end for
    end if
    if dist[b][t'] = ℓ then
        pre[b][t'].add((a, t))
    end if
end function
function Relax_extend(b, t', dist, pre, ℓ)
    if dist[b][t'] > ℓ then
        dist[b][t'] = ℓ, pre[b][t'] = {}
    end if
end function
```

Proposition 3 allows to compute the values of $\delta_{s\cdot}$ by recurrence for all temporal nodes by starting the recurrence from the sources of G_s.

Algorithm 2 Computes the values of $B(v, t)$ for all temporal nodes

Input: $G = (V, \mathcal{E}, T)$: a temporal graph, k : maximal waiting time for restless walks or $k = \infty$ for shortest walks.
Output: $B(v, t), \forall v \in V, t \in [T]$

function $\textsc{Betweenness\_centrality}(G, k)$
  $B(v, t) = 0, \forall v \in V, t \in [T]$
  for $s \in V$ **do**
    $pre, d = \textsc{Temporal\_BFS}(G, s, k)$ ▷ Algorithm 1
    $del, sig = \textsc{count\_walks}(pre, d, s)$ ▷ Apply Corollary 1
    $cum = \textsc{Contributions}(G, pre, del, sig)$ ▷ see Algorithm 3
    $\textsc{Update\_betweenness}(B, cum, del)$ ▷ Apply Equation (4)
  end for
  return B
end function

Optimal active walks

We keep the same notations as in the previous section. We denote by $W^{act}_{s(v,t)}$ the set of active $s - (v, t)$ walks. For the active case we will have to consider two types of walks, either walks where the last transition of the walk is (w, v, t) which we call an exact-$s - (v, t)$ walk or the walk arrived earlier to v at time $t' < t$. We recall that in this we only show that active shortest walks can be computed efficiently that is $k = \infty$ and therefore we drop k all together from the notation. Then:

$$\begin{aligned} W^{act}_{s(v,t)} =& \{W | W \in W_{sv}, arr(W) \leq t\} \\ & \cup \{\epsilon \mid s = v, (s, w, t) \in \mathcal{E} \text{ for some } w \in V\}. \end{aligned}$$

Equations (5), (6), (7) generalize immediately to the active case. We define $c^{act}_s(v, t)$ and see that we can as in the previous section compute the values of Eq. (1) as follows:

$$c^{act}_s(v, t) = \min_{W \in W^{act}_{s(v,t)}} (c^{sh}(W)), \quad c^{sh}_s(v) = \min_{t \in [T]} (c^{act}_s(v, t)). \tag{11}$$

If the set of $s - v$ walks is empty then $c^{act}_s(v, t) = \infty$. Table 2 shows the c^{act}_a values of the graph of Fig. 1. Then we can define the set of optimal active $s - (v, t)$ walks:

$$\mathcal{W}^{act}_{s(v,t)} = \{W | W \in W^{act}_{s(v,t)}, c(W) = c^{act}_s(v, t)\}. \tag{12}$$

From our definition we see that $\mathcal{W}^{act}_{s,(v,t)}$ and $\mathcal{W}^{\infty,pas}_{s,(v,t)}$ do not coincide in general. For instance, take $W = a \xrightarrow{1} b \xrightarrow{2} c$, then $W \in \mathcal{W}^{act}_{a,(c,5)}$ since W arrives to node c at time 2 while $W \notin \mathcal{W}^{\infty,pas}_{a,(c,5)}$ since it is not an exact walk. On the other hand $W' = a \xrightarrow{1} b \xrightarrow{2} c \xrightarrow{5} b$ is such that $W' \in \mathcal{W}^{\infty,pas}_{a,(b,5)}$ while $W' \notin \mathcal{W}^{act}_{a,(b,5)}$ since in the

Table 2 Values of $c^{\infty,pas}_s(v, t)$, $c^{\infty,act}_s(v, t)$ and $c^{sh}_s(v)$ on the temporal graph of Fig. 1

t	0	1	2	3	4	5	6
$c^{\infty,pas}_a(b, t)$	∞	1	∞	∞	∞	3	∞
$c^{\infty,act}_a(b, t)$	∞	1	1	1	1	1	1
v	a	b	c	d	e		
$c^{sh-\infty}_a(v)$	0	1	2	3	∞		

(Upper part) values of $c^{\infty,pas}_a(b, t)$ of node b at different times t for passive walks (1st row), and $c^{\infty,act}_a(b, t)$ for active walks (2nd row) and for all $t \in [T]$. (Lower part) Overall optimal values of $c^{sh-\infty}_a(v)$ (third row) for all $v \in V$

passive case the optimal cost from a to $(b,5)$ is 3 while in the active case this cost is 1 see Table 2. We also see that in the active case, a walk W can be optimal to different times on v. For instance on the graph of Fig. 1. The walk $W = a \xrightarrow{1} b \xrightarrow{2} c$, W is an optimal $a-(c,2)$ walk $W \in \mathcal{W}^{act}_{a,(c,2)}$ and W is also an optimal $a-(c,5)$ walk $W \in \mathcal{W}^{act}_{a,(c,5)}$.

The predecessor graph presented in Definition 8 extends directly by using the set elements from the set $\mathcal{W}^{act}_{s(v,t)}$. We have:

Definition 10 *(predecessor set, successor set)* Let $G=(V,\mathcal{E},T)$ be a temporal graph and fix a source $s \in V$, then for all $w \in V, w \neq s$, let $\mathcal{W}^{act}_{s,(v,t)}$ be the set of optimal $s-(v,t)$ walks:

$$\begin{aligned} pre_s(w,t') =& \{(v,t) \in V \times [T] \mid \exists\, m \in \mathcal{W}^{act}_{s(w,t')}, m = s \xrightarrow{t_1} \ldots \xrightarrow{t} v \xrightarrow{t'} w\} \\ & \cup \{(s,t') \mid \exists\, m \in \mathcal{W}^{act}_{s(w,t')}, m = s \xrightarrow{t'} w\}. \end{aligned}$$

The successor set of a node is defined similarly $succ_s(w,t') = \{(v,t) \mid (w,t') \in pre_s(v,t)\}$.

Moreover, the predecessor graph in the case of active walks is a subgraph of the predecessor graph from the passive case as will be shown in the next proposition. An example of the predecessor graph for active shortest walks is given on Fig. 2.

Proposition 4 *Given a temporal graph $G=(V,E,T)$ and $s \in V$. Let $G^{pas}_s = \left(V^{pas}_s, E^{pas}_s\right)$ be the predecessor graph from s in the shortest passive case and let $G^{act}_s = \left(V^{act}_s, E^{act}_s\right)$ be the predecessor graph in the shortest active case with $k=\infty$. Then, $E^{act}_s \subseteq E^{pas}_s$.*

Definition 11 ($Exact\ optimal\ s-(v,t)\ walks$) We say that an $s-(v,t)$ walk W is an *exact optimal* $s-(v,t)$ *walk* if $W \in \mathcal{W}^{act}_{s(v,t)}$ and $arr(W)=t$. We denote by $\overline{\mathcal{W}}^{act}_{s(v,t)}$ the set of all *exact optimal* $s-(v,t)$ *walks*. We define $\overline{\sigma}_{s,(v,t)} = |\overline{\mathcal{W}}^{act}_{s,(v,t)}|$ which corresponds to the number of optimal exact $s-(v,t)$ walks.

Proposition 5 *For any temporal node (v,t), there holds that:*

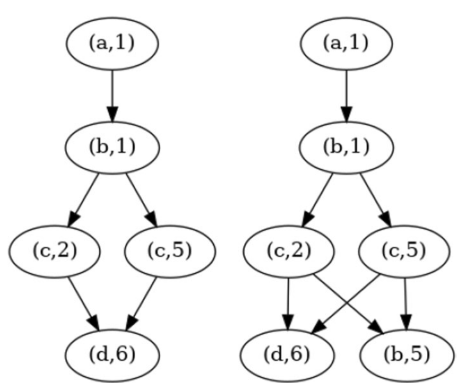

Fig. 2 The predecessor graphs of shortest paths from node *a* on the temporal graph of Fig. 1. (left) the walks are considered active and (right) the walks are considered passive

$$\overline{\sigma}_{s(v,t)} = \begin{cases} 0 & \text{if } (v,t) \neq V_s, \\ 1 & \text{if } (v,t) \in V_s \text{ } and \text{ } v = s, \\ \sum\limits_{(w,t') \in pre_s(v,t)} \overline{\sigma}_{s(w,t')} & \text{otherwise.} \end{cases} \tag{13}$$

Now defining $\sigma_{s(v,t)} = |\mathcal{W}^{act}_{s,(v,t)}|$ we show that:

Proposition 6 *Let $s \in V$, $G_s = (V_s, E_s)$ be the predecessor graph from node s and for any temporal node $(v,t) \in V \times [T]$:*

$$\sigma_{s(v,t)} = \sum_{\substack{t' \leq t \\ c_s^{act}(v,t) = c_s^{act}(v,t')}} \overline{\sigma}_{s(v,t')}, \tag{14}$$

Finally the quantities σ_{sv} and $\sigma_{sv}(v,t)$ for all $s, v \in V$ and $t \in [T]$ can be computed in the same way using Eq. (9):

$$\sigma_{sv} = \sum_{\substack{t \in [T] \\ c_s^{act}(v,t) = c_s^{sh}(v)}} \overline{\sigma}_{s(v,t)}, \quad \sigma_{sv}(v,t) = \begin{cases} \sigma_{s(v,t)} & \text{if } c_s^{act}(v,t) = c_s^{sh}(v) \\ 0 & \text{otherwise.} \end{cases} \tag{15}$$

We first define $\mathrm{bef}_{\mathrm{G_s}}(\mathrm{v,t})$ to be the largest time t' such that $t' \leq t$ and $(v,t') \in V_s$. Therefore if $(v,t) \in V_s$, then $\mathrm{bef}_{\mathrm{G_s}}(\mathrm{v,t}) = \mathrm{t}$. Then Eq. (10) can be extended to the active case with a slight modification:

Proposition 7 *Fix a node $s \in V$, considering active walks then for $(v,t) \in V_s$*

$$\delta_{s\cdot}(v,t) = \delta_{sv}(v,t) + \sum_{\substack{t'' := \mathrm{bef}_{\mathrm{G_s}}(\mathrm{v,t}) \\ (w,t') \in succ_s(v,t'') \\ t' \geq t}} \frac{\sigma_{s(v,t)}}{\sigma_{s(w,t')}} \delta_{s\cdot}(w,t'), \tag{16}$$

For active walks, we finally need to ensure that the general contribution is also computed for temporal nodes not lying on the predecessor graph. We discuss this in the Appendix and show that the values of these temporal nodes can be computed on the fly during the general contribution recurrence provided that we order the predecessor graph which adds a factor in the final complexity of active walks.

Finally, in Sect. 5 it will become clear why active k-restless walks for $k < \infty$ does not hold. This is a consequence of Eq. (16) not being true in that case.

Computation details for passive and active walks

We use a temporal BFS algorithm variant of the one used in Buß et al. (2020). The relaxing technique builds the shortest $s-(v,t)$ walks and for active walks it checks if the extension of a walk arriving to (w,t') is also shortest to (w,t'') with $t'' > t'$. The procedure is defined in function RELAX of Algorithm 1.

Definition 12 *(Exactly reachable temporal nodes)* Let $G = (V, \mathcal{E}, T)$ be a temporal graph and fix a source $s \in V$. Then we define:

$$ER_s = \{(v,t) \mid (v,t) \in V \times [T], \exists k\text{-restless walk } W = s \xrightarrow{t_1} \dots \xrightarrow{t} v\}.$$

In essence, ER_s keeps all temporal nodes that are endpoints of some temporal walk.

Proposition 8 *Algorithm* TEMPORAL_BFS *solves the optimal walk problem for a temporal graph* $G = (V, \mathcal{E}, T)$. *That is for all* $(v,t) \in ER_s$, $dist[v][t] = c_s(v,t)$, $pre[v][t] = pre_s(v,t)$ *and* (v,t) *is added exactly once in the queue* Q.

Proposition 9 *Let* $G = (V, \mathcal{E}, T)$ *be a temporal graph, fix a walk type and a source* $s \in V$, *then the predecessor graph* G_s *can be computed in* $O(mT + nT)$.

Corollary 1 *For both walk types and for all* $v \in V$ *and* $t \in [T]$ *the quantities* $\sigma_{s(v,t)}$ *and* $\delta_{sv}(v,t)$ *can be computed for all temporal nodes* (v,t) *in* $O(m\,T + n\,T)$.

Finally, to compute the betweenness centrality of the whole temporal graph it suffices to sum $\delta_{s\cdot}$ for all $s \in V$ and use the correction formula given in Eq. (4) needed to go from $\hat{B}(v,t)$ to $B(v,t)$. These steps are summarised in Algorithm 2. In the Algorithm function COUNT_WALKS:

- To compute $\sigma_{s(v,t)}$ for all $v \in V$ and $t \in [T]$:
  - (passive case) Applies Eq. (8)
  - (active case) Applies Eq. (13) followed by Eq. (14)
- To compute $\sigma_{sv}, \sigma_{sv}(v,t)$ for all $v \in V$ and $t \in [T]$:
  - (passive case) Applies Eq. (9)
  - (active case) Applies Eq. (15)

Finally, function COUNT_WALKS returns a dictionary *del* containing the values of $\delta_{sv}(v,t)$ for all $v \in V$ and $t \in [T]$ and another dictionary *sig* containing the values of $\sigma_{s(v,t)}$ for all $v \in V$ and $t \in [T]$.

Proof of Theorem 1 For active and passive walks, the total cost of the predecessor graph construction in (Line 4) of Algorithm 2 is for any node $s \in V$ in $O((n+m)T)$. Then we can compute all necessary quantities for the main recurrence (Line 5). This is also done in $O(mT + nT)$ as well as explained in Corollary 1. The recurrence of Proposition 3 computing all contributions from node $s \in V$ (Line 6) can be computed in $O(mT + nT)$ for passive walks. However, for active walks the same computations can be done and by ordering the predecessor graph we can compute contributions of temporal nodes not lying on the predecessor graph (see discussion in Appendix 5). The predecessor graph ordering costs then $O(nT^2)$ and the overall cost of (Line 6) is $O(mT + nT^2)$ for active walks. Finally, the application of the correction formula in Line 7 can be done in $O(nT)$.

□

Table 3 Statistics for the datasets

Dataset	Nodes	Events	Edges	agg_edges	Buss	sh pas	sh act	sh-rl pas	sh-fm pas	Static
gre	1547	1300	113112	3680	1924.1	2921.2	9067.7	1496.3	3839.8	21.156
ren	1407	10825	107384	3718	2480.7	3343.4	26144	1656.3	3329.3	29.046
bel	1917	1132	120951	6440	2530.8	3512.9	12691.	2056.5	3753.4	45.057
kuo	549	1211	30545	1952	169.45	242.67	683.99	119.67	240.69	1.9935
prim	242	3100	125773	16634	1146.2	1701.6	454.91	761.39	1597.1	4.0661
hs11	126	5609	28539	3418	75.741	98.333	101.61	40.064	107.29	0.3405
hs12	180	11273	45047	4440	208.56	287.82	371.63	142.94	299.61	0.9211
hp	75	9453	32424	2278	121.75	170.15	76.980	75.908	177.97	0.1877
ht	113	5246	20818	4392	43.157	61.687	63.644	34.362	66.632	0.2166
wp	92	7104	9827	1510	12.020	16.154	51.370	7.4984	17.592	0.0782

From left to right number of nodes (nodes), number of times (events), number of temporal edges (edges), number of edges in the static representation of *G*. The execution time in seconds of (Buß) implementation of Buß et al. (2020), Algorithm 2 active shortest (sh act), passive shortest (sh pas), passive shortest restless (sh-rl pas), passive shortest foremost (sh-fm) and on the aggregated static graph of *G*

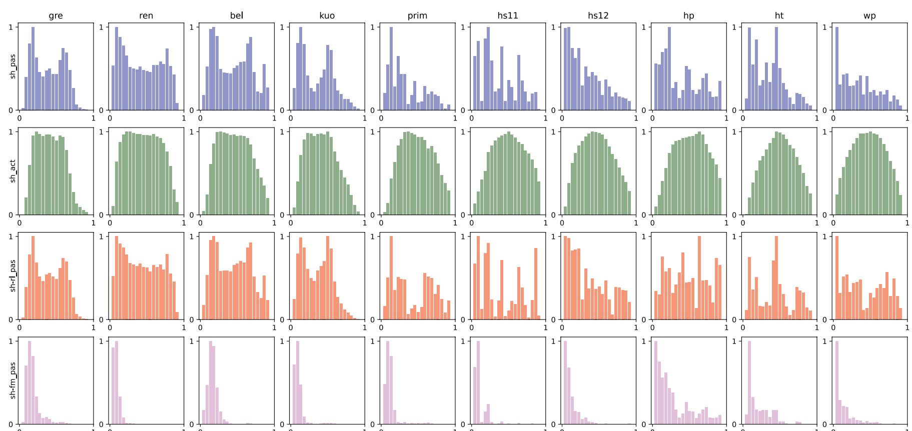

Fig. 3 Distribution of time centrality values *B*(*t*) for the datasets. Each column represents a dataset. (1st row) correspond to the distribution of *B*(*t*) for passive shortest, (2nd row) active shortest, (3rd row) passive shortest restless and (4th row) passive shortest foremost. The x-axis represents the renormalized life time of the temporal graph and the y-axis represents the values of *B*(*t*) grouped into 20 bars

Experimental results

For our experiments we built our algorithms on top of the code of Buß et al. (2020). We implemented our algorithms with the different possible variants. We focus next on the variants of active and passive shortest walks, passive shortest k-restless walks for k being equal to 10% of the lifetime of the graph and passive shortest foremost walks.

We summarize here our main findings:

- The active variants takes more time to be computed than the passive one as stated by Theorem 1 but this difference starts to emerge with large networks. See Table 3.
- On the importance of times $B(t)$, the active variant points the importance of central times in comparison with the passive variant (Fig. 3).
- The rankings between (passive shortest), (active shortest) and the betweenness centrality computed on the aggregated static graph are positively correlated. There

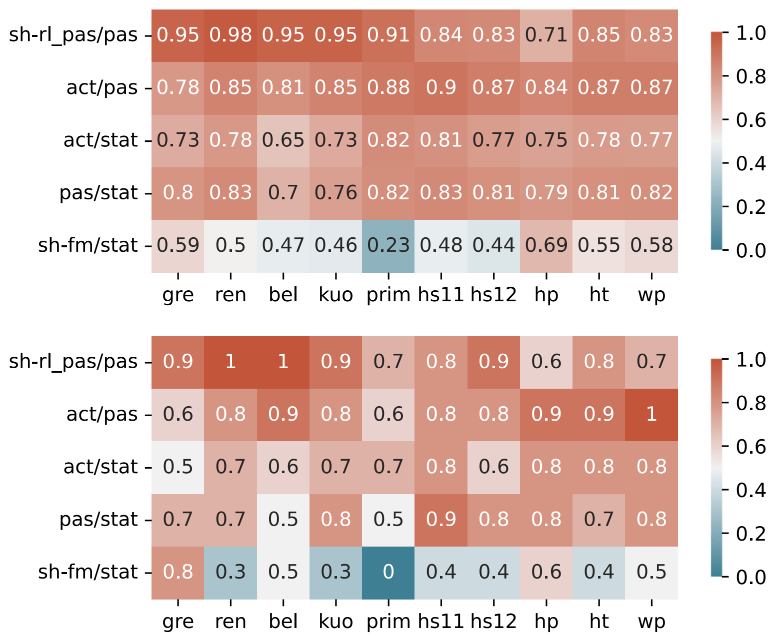

Fig. 4 Heatmap of betweenness centrality of *B*(*v*) comparisons of the datasets. (left) Kendall-tau rank correlation rankings and (right) intersection rate of the top 20 nodes. act stands for active variant, pas for passive variant, sh-rl_pas passive shortest restless and stat for the static betweenness centrality on the aggregated graph

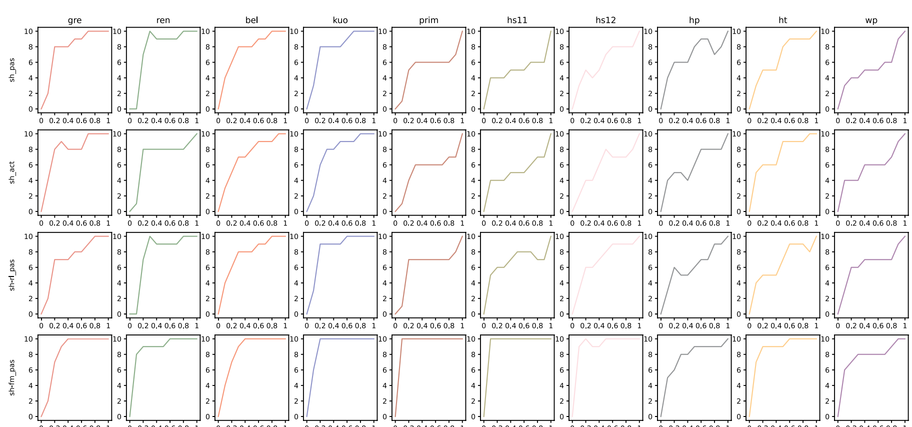

Fig. 5 Each column corresponds to a dataset. (1st row) correspond to the distribution of *B*(*t*) for passive shortest, (2nd row) active shortest, (3rd row) passive shortest restless and (4th row) passive shortest foremost. Each graph has on its x-axis the μ values and on its y-axis the size of the intersection between top 10 ranked nodes of nodes $B_G(v)$ of the temporal graph G and top 10 ranked nodes of the $B_{G\leq\mu}(v)$

are more differences between these variants when looking at the intersection of the top ranked 10-nodes (Fig. 4).

- Predicting the top 10 ranked nodes by looking only at the first few interactions is much more accurate for passive shortest foremost variant compared to the others (Fig. 5).

Our code is open-source[1] and is written in C++. We used an Intel(R) Xeon(R) Silver 4210*R* CPU 2.40GHz without parallel processes. The datasets are divided into two types. Public transport datasets Kujala et al. (2018). The datesets are: gre (grenoble), ren (rennes), bel (belfast) and kuo (kuopio) and social contact traces from http://www.sociopatterns.org/ namely: prim (primaryschool), hs11 (HighSchool2011), hs12 (HighSchool2012), hp (Hospital Ward), ht (HyperText) and wp (Workplace). All datasets are available publicly.

On table 3 we give information about the datasets that we used as well as running times of our algorithms and the one of Buß et al. (2020). Our implementation is complementary to theirs since the authors of Buß et al. (2020) compute the overall betweenness centrality of nodes $B(v)$ for passive shortest walks. In comparison, our implementation computes all the values of $B(v, t)$, $B(v)$ and $B(t)$ for both active and passive variants. We note that according to Theorem 1 the active version is slower than its passive counterpart and the difference becomes more clear on larger graphs while on smaller networks the execution times are more comparable and for some instances the active version is faster than the passive one. This happens whenever the overall cost of ordering the predecessor graphs is less important than the overall gain in the sizes of the predecessor graphs which are smaller in the active case, see Proposition 4.

On Fig. 3, we see that the distribution of the values of $B(t)$ are much more concentrated around central times (those in the middle of the temporal graph) for the shortest active walks than for shortest passive walks due to the contribution of intermediary node (first two rows) showing the difference between active and passive walks. For passive shortest walks important times tend to be in the beginning of the lifetime of the graph since many shortest walks can be formed when starting walks early in time and combining times later on. Finally, for passive foremost variant (last row) the most important times are seen in the beginning of the graph due to the fact that this measure is focused on temporal walks arriving the earliest.

We compared the ranking correlation and intersection size of the different proposed variants together with the static betweenness centrality on the aggregated graph gives. We also compared passive shortest and passive shortest k-restless variants together. On Fig. 4 we see that the rankings and intersection of top 20 nodes of $B(v)$ for several real world datasets show that passive betweenness centrality and the static one have a high correlation for all our datasets, always higher than the correlation between the active and static one (rows 4 and 5). Our results suggest that if we want an approximate ranking of $B(v)$ for passive (and to a lesser extent active) shortest walks, the static betweenness centrality gives a good approximation to it and runs much faster (100 times faster for all our datasets) than the temporal version. While the comparison between active temporal betweenness centrality and the static one show less correlation in general. The comparison between active and passive variants shows high correlation for $B(v)$ while the behaviour is largely different for $B(t)$ between active and passive variants. If we care about arriving first to nodes (passive shortest foremost version) we see that the

[1] https://github.com/busyweaver/code/_temporal/_betweenness/_

correlations with the betweenness on the static aggregated graph has a low correlation (last row).

Finally, in many practical applications we have access only to the first few interactions of the graph but we still want to predict the node rankings. Here we focus on predicting the overall node ranking $B(v)$ rather than the temporal one $B(v, t)$ - in fact it has been argued in a close context (closeness centrality) that predicting the temporal node centrality evolution is in general difficult Magnien and Tarissan (2015) since distances can vary a lot by the addition of one new temporal edge. In order to do so we introduce the graph of first times as follows:

Definition 13 (*The graph of first μ times of G*) Let $G = (V, \mathcal{E}, T)$ be a temporal graph and let $\mu \in [0, 1]$, and let then $G^{\leq\mu} = (V, \mathcal{E}', \mu T)$ with $\mathcal{E}' = \{(u, v, t) \mid (u, v, t) \in \mathcal{E}, t \leq \mu T\}$.

Our results are summarized in Fig. 5 in which each plot corresponds to how well are the top 10 nodes ranked when looking only at the first times of the graph. Therefore, the faster each plot reaches 10 the fewer interactions we need to observe to correctly identify the top 10 ranked nodes. It turns out that this depends on the criteria that we consider. For instance, considering passive shortest foremost variant (3rd row) looking at the first 10% interactions gives a good approximation of the top ranked nodes. In fact this is in accordance with the last row of Fig. 3 where we see that most important times $B(t)$ in the graphs are the ones in the beginning of it.

Details of the proofs

A path in the predecessor graph will be denoted denoted $(s, t_1) \to (v_1, t_1) \to \cdots \to (v_n, t_n)$. We also use the operator $\oplus$ for the concatenation of walks.

Proof of Lemma 1 By induction we show that a path in the predecessor graph G_s starting at node s corresponds to an $s - (v, t)$ walk. Let p be a path in the predecessor graph. Then $p \equiv p' \to (v, t)$ for some path p'. Let (w, t') be the last node in p'. By induction hypothesis, suppose that p' corresponds to an optimal $s - (w, t')$ walk P'. Since $((w, t'), (v, t)) \in E_s$, this implies that there exists an optimal $s - (v, t)$ walk M that passes through (w, t') before arriving to (v, t). Let $M \equiv M' \oplus (w \xrightarrow{t} v)$. M' is necessarily an optimal $s - (w, t')$ walk since else M would not be an optimal $s - (v, t)$ walk. Now M' and P' are both optimal $s - (w, t')$ walks implying that they have the same length. Since M extends M' with a single edge then $c_s(v, t) = c_s(w, t') + 1$. Therefore, $P \equiv P' \oplus (w \xrightarrow{t} v)$ is an optimal $s - (v, t)$ walk as well. On the other hand, We show by induction that an exact optimal $s - (v, t)$ walk P corresponds to a single path in G_s starting at (s, i) for some time i. Let (w, t') be the last node appearance before (v, t). Then $P \equiv P' \oplus (w \xrightarrow{t} v)$. P' is an optimal $s - (w, t')$ walk. Then by induction hypothesis, let p' be the corresponding path in predecessor graph. Then $((w, t'), (v, t)) \in E_s$ and the path $p \equiv p' \to (v, t)$ is a path in the predecessor graph. □

Proof of Lemma 2 Using Lemma 1, there are no exact optimal $s-(v,t)$ walks containing cycles because else we could immediately construct an exact $s-(v,t)$ walk W without cycles and $len(W) < c_s(v,t)$ which is impossible. Hence the predecessor graph from s is a acyclic. $\square$

Proof of Proposition 2 We know that $\sigma_{s(v,t)}$ corresponds to the number of walks in the predecessor graph from s ending in (v,t) by Lemma 1. By induction any optimal $s-(v,t)$ walk comes from a predecessor (w,t') with $(w,t') \in pre_s(v,t)$. Each optimal $s-(w,t')$ walk can be extended uniquely by appending $(w \xrightarrow{t'} v)$ to it and make it an optimal exact $s-(v,t)$ walk. Finally, Lemma 2 ensures that the recurrence is well founded. $\square$

Proof of Proposition 4 Let us denote by $p_s(v,t)$ the predecessor set of $\mathcal{W}^{\infty,pas}_{s(v,t)}$ from node $s \in V$ to the temporal node (v,t). We also denote by $q_s(v,t)$ the predecessor set of $\mathcal{W}^{act}_{s(v,t)}$. Then, for all $s,v \in V, t \in [T]$ we have $q_s(v,t) \subseteq p_s(v,t)$. let $(w,t') \in q_s(v,t)$, then there exists an optimal $s-(v,t)$ walk arriving to w at t' and then taking the edge (w,v,t). By definition $W \in \mathcal{W}^{act}_{s,(v,t)}$, but then we also have that $W \in \mathcal{W}^{\infty,pas}_{s,(v,t)}$ since W ends exactly at t and it is a shortest walk. Therefore, $(w,t') \in p_s(v,t)$. We will also show that There exists a graph and nodes $s,v \in V$ and $t \in [T]$ such that $p_s(v,t) \not\subseteq q_s(v,t)$. Take the graph of Fig. 1. Then $p_a(b,4) = \{(c,2)\}$ since $W = a \xrightarrow{1} b \xrightarrow{2} c \xrightarrow{4} b$ is such that $W \in \mathcal{W}^{\infty,pas}_{a,(b,4)}$ while $W \notin \mathcal{W}^{act}_{a,(b,4)}$ since $\mathcal{W}^{act}_{a,(b,4)}$ only contains $W' = a \xrightarrow{1} b$ and therefore $q_a(b,4) = \emptyset$. $\square$

Definition 14 *(arc dependency)* Fix a node $s \in V$ and a type of walks. Then, $\delta_{sz}(v,t,(v,w,t')$ denotes the fraction of optimal $s-z$-walks in $\mathcal{W}$ that go through the node appearance (v,t) and then use the temporal arc $(v,w,t') \in \mathcal{E}$.

Lemma 3 *Let $G=(V,\mathcal{E},T)$ be a temporal graph, fix $k \in \mathbb{N}$ and a node $s \in V$. Let $G_s = (V_s, E_s)$ be the predecessor graph from s. Suppose either passive case (any k) or active case ($k=\infty$). Let (v,t) be a temporal node and $(v,w,t') \in \mathcal{E}$. If $\delta_{sz}(v,t,(v,w,t')) > 0$, then*

$$\delta_{sz}(v,t,(v,w,t')) = \frac{\sigma_{s(v,t)}}{\sigma_{s(w,t')}} \frac{\sigma_{sz}(w,t')}{\sigma_{sz}}.$$

Proof of Lemma 3 For passive walks, only temporal nodes $(v,t) \in V_s$ can have strictly positive values of $\delta_{sz}(v,t,(v,w,t'))$. The proof can be generalized from Lemma 5.2 in Rymar et al. (2021) by noticing that the fraction $\frac{\sigma_{sz}(w,t')}{\sigma_{s(w,t')}}$ corresponds to the number of k-restless optimal walk suffixes starting at (w,t') and ending in z. Multiplying this by $\sigma_{s(v,t)}$ makes all optimal $s-(v,t)$ walks. We know that any walk represented by $\sigma_{s(v,t)}$ is k-restless. So we are left with showing that $t'-t \leq k$. We know all walks in this case visit (v,t) before taking the edge (v,w,t') and since $\delta_{sz}(v,t,(v,w,t')) > 0$ it means that there exists an optimal k-restless walk visiting (v,t) and taking the edge (v,w,t'). Therefore $t'-t \leq k$, thus, the walks are entirely k-restless.

For active walks the fraction $\frac{\sigma_{sz}(w,t')}{\sigma_{s(w,t')}}$ also corresponds to the number of optimal k-restless suffixes starting at (w,t') and ending in z. Multiplying this by $\sigma_{s(v,t)}$ makes all optimal $s-(v,t)$ walks. However, since $\sigma_{s(v,t)}$ can contain optimal walks that arrived before t. We can not ensure anymore that the product ensures that $(t'-arr(W))\leq k$ for all optimal $s-(v,t)$ k-restless W walks. Thus, this case is only true when $k=\infty$.

Finally, if in Definition 3, we would not have extended the optimal walks to W_T, this fraction would not correspond to the suffixes when $z=w$. Apart from this detail the rest of the proof also follows from the same reference. □

Remark 3 In the Proof of Lemma 3, we showed that optimal $s-z$-walks in $\mathcal{W}$ that go through the node appearance $(v,\ t)$ and then use the temporal arc (v,w,t') can be constructed using optimal $s-(v,t)$ and optimal $s-(w,t')$ walks. In the passive case we know exactly when these walks end and therefore we can ensure that the resulting walks respects the k-restlessness. In the active case however optimal $s-(v,t)$ can arrive to v at $t''<t$ and therefore it is not possible to ensure k-restlessness in this scheme. This is why our scheme only works in the active case for $k=\infty$.

Proof of Proposition 3 The proofs closely follows the one of Lemma 5.3 in Rymar et al. (2021) by using Lemma 3. □

Proof of Proposition 5 The same results of Lemmas 1 and 2 can be shown (using the same arguments) on the predecessor graph in the active case as well using exact optimal walks. Finally the optimal $s-(v,t)$ walks in $\mathcal{W}^{act}_{s(v,t)}$ have the prefix property of $\mathcal{W}^{\infty,pas}_{s(v,t)}$ and the same proof of Proposition 2 holds. □

Proposition 6 *Recall that $\mathcal{W}^{act}_{s,(v,t)}$ contains optimal walks arriving exactly at t or walks arriving earlier. Therefore, for any $t'\leq t$ if $c^{act}_s(v,t)=c^{act}_s(v,t')$ all the walks arriving exactly at t' count in $\sigma^{act}_{s(v,t)}$.* □

Proposition 7 *In the active case with $k=\infty$ we notice that $(v,\ t)$ might not belong to V_s. Therefore all shortest paths passing through $(v,\ t)$ will pass through the first $t''<t$ with $(v,t'')\in V_s$. Hence, the index of the sum looks at the successors of (v,t'') in G_s and only need to consider those successors (w,t') with $t'\geq t$ so that these paths pass through (v,t'') then $(v,\ t)$ and then go to (w,t'). The rest of the proof follows the one of Lemma 5.3 in Rymar et al. (2021).* □

Proof of Proposition 8 We show that at the start of the i-th iteration of the loop at Line 4 in Algorithm 1, all temporal nodes $(v,\ t)$ which are exactly reachable with $i-1$ edges have the four following properties: $\mathrm{dist[v][t]=c_s(v,t)=i-1}$, $(v,t)\in Q$, $pre[v][t]=pre_s(v,t)$ and $(v,\ t)$ is added exactly once to the queue. We denote by Q_i the queue Q at the start of the i-th iteration. The property is true just before entering the loop the first time. Only temporal nodes $(s,\ t)$ where $(s,w,t)\in\mathcal{E}$ for some $w\in V$ are in the queue. The condition (not $((s=a)$ and $(t'\neq t)))$ can only be met during the first iteration of the main loop. It ensure that all paths of length 1 created have their first appearance time at the time of their first transition. Suppose that the properties hold for the start of iteration $i-1$. Let (w,t') be an exactly reachable temporal node with $c_s(w,t')=i$. Then, the shortest $s-(w,t')$ walk is of length i. There are two possible cases, either there exists an exact walk $s-(w,t')$ walk with a last edge of the form (v,w,t') for some $v \in V$ or all shortest $s-(w,t')$ walks arrive at times $t'' < t'$ (can only happen in active setting). Suppose that there exists an exact shortest walk $W = s \overset{t_1}{\rightarrow} \ldots \overset{t_{i-1}}{\rightarrow} v \overset{t_i}{\rightarrow} w$ with $t_i = t'$ and $t_{i-1} = t$. Now, take the walk W' which corresponds to the walk W without its last transition. Since W is a shortest $s-(w,t')$ walk then W' is necessarily a shortest $s-(v,t)$ walk and then $c_s(v,t) = i-1$ and therefore by induction hypothesis $(v,t) \in Q_{i-1}$. During the $i-1$-th iteration, $(v,\ t)$ was treated and the edge (v,w,t') was relaxed. Now either (w,t') was reached previously during the this same iteration by the relaxation of some edge (f,w,t') for some $f \in V$ in which case $dist[w][t']$ is already equal to i and (w,t') is already in Q_i. If (w,t') was never reached during this iteration, then $dist[w][t']$ is set to i and (w,t') is added to Q_i. Note that (w,t') is added only once to Q_i during this iteration due to the condition on the first line of the RELAX_EXTEND function. In the second case, there exists an exact shortest walk $W = s \overset{t_1}{\rightarrow} \ \ldots \overset{t_{i-1}}{\rightarrow} v \overset{t_i}{\rightarrow} w$ with $t_i = t''$, $t'' < t'$ and $t_{i-1} = t$. In this case RELAX_EXTEND is called on (w,t') when relaxing (w,t''), $dist[w][t']$ is set to i and $pre[w][t']$ is set to empty which corresponds to the fact that there are no exact shortest walks to (w,t'). Finally, since ℓ increases at each iteration (w,t') is never added again to any Q since $dist[w][t']$ is always strictly smaller than subsequent values of ℓ. Finally, let $(v,t) \in pre_s(w,t')$, then necessarily $(v,t) \in Q_{i-1}$ and hence will be added to $pre[w][t']$ in the RELAX_EXTEND function. Therefore, at then end of the $i-1$-th iteration $pre[w][t'] = pre_s(w,t')$. □

Proof of Proposition 9 By using a queue each temporal node $(v,t) \in ER_s$ is scanned at most one time by TEMPORAL_BFS of Algorithm 1 as it was shown in Proposition 8. Then the same temporal arc $(v,w,t) \in \mathcal{E}$, can be relaxed up to T times in Line 4 of Algorithm 1 and T other times in Line 5 in function RELAX of Algorithm 1. Remember that each temporal nodes in ER_s is added at most once to the queue. Thus the overall time complexity of Algorithm 1 is $O(m\,T + n\,T)$. □

Proof of Corollary 1 For passive walks, $\sigma_{s(v,t)}$ can be computed recursively from the predecessor graph using Eq. (8). Then σ_{sz} and $\sigma_{sv}(v,t)$ can be computed using Eq. (9). For active walks $\overline{\sigma}_{s(v,t)}$ can be computed recursively from the predecessor graph using Eq. (13) and then $\sigma_{s(v,t)}$ can be computed using Eq. (14). σ_{sz} and $\sigma_{sv}(v,t)$ can be computed using Eq. (15). □

General contributions for temporal nodes not lying on the predecessor graph

In the active case, there are temporal nodes that have betweenness centrality values from node s without being on the predecessor graph of s. This can not happen in the passive case. For instance, on the temporal graph of Fig. 1 the temporal node $(b,\ 3)$ has no contribution for the walk W if it is considered passive (red path) but it has contribution if W is considered active (blue path).

Recall that $\mathrm{bef}_{G_s}(v,t)$ is be the largest time t' such that $t' \leq t$ and $(v,t') \in V_s$. Let $G_s = (V_s, E_s)$ be the predecessor graph from s. Let $t' = \mathrm{bef}_{G_s}(v,t)$. Then $\sigma_{s(v,t)} = \sigma_{s(v,t')}$. This result can be seen since we know that $t' \leq t$, and $(v,t') \in V_s$. If $t' = t$ the result is immediate. If $t' < t$, then $(v,t) \notin V_s$, and therefore there are no exact optimal $s-(v,t)$ walks. All the optimal $s-(v,t)$ walks are those arriving from t'. As a consequence we saw in Eq. (16) that:

$$\delta_{s\cdot}(v,t) = \delta_{sv}(v,t) + \sum_{\substack{t'' := \mathrm{bef}_{G_s}(v,t) \\ (w,t') \in succ_s(v,t'') \\ t' \geq t}} \frac{\sigma_{s(v,t'')}}{\sigma_{s(w,t'')}} \delta_{s\cdot}(w,t'),$$

This last Equation ensures that for active walks the computation $\delta^{act}_{s\cdot}(v,t)$ for temporal nodes $(v,t) \notin V_s$ can be done on the fly while computing $\delta^{act}_{s\cdot}(v,t')$ with $t' = \mathrm{bef}_{G_s}(v,t)$. This implies computing the elements (w,t'') of the successor set of (v, t) in a decreasing order of time and give the value $\delta^{act}_{s\cdot}(v,t)$ before the sum has completed since we need to stop when the elements (w,t'') have $t'' < t$. Algorithm 3 shows this part in the function INTER_CONTRIBUTION.

As an example we compute the contributions from node a to the temporal node $(b, 3)$ on the temporal graph of Fig. 1. We see that $(b, 3)$ does not belong to the predecessor graph from a $G_a = (V_a, E_a)$ $(b,3) \notin V_a$ (the left graph of Fig. 2). We have that $\mathrm{bef}(b,3) = (b,1)$, $succ_a(b,1) = \{((c,2),(c,5)\}$ Its contribution $\delta_{a\cdot}(b,2) = \frac{\sigma_{a(b,1)}}{\sigma_{a(c,5)}}\delta_{a\cdot}(c,5) = \frac{1}{1}(1+1) = 2$. In the passive case temporal node $(b, 3)$ has no contributions from any node and therefore $\delta_{a\cdot}(b,2) = 0$.

Algorithm 3 Compute the values of $\delta_{s\cdot}(v,t) = 0$ for a temporal graph G

Input: $G = (V, \mathcal{E}, T)$: a temporal graph, $G_s = (V_s, E_s)$ the predecessor graph of s, del the values of $\delta sv(v,t)$ for all (v,t) and sig the values of $\sigma_{s(v,t)}$ for all (v,t)
Output: A dictionary cum containing the values of $cum[(v,t)] = \delta_{s\bullet}(v,t), \forall v \in V, t \in [T]$

```
function CONTRIBUTIONS(G, G_s, del, sig)
    cum(v,t) = 0, ∀v ∈ V, t ∈ [T]
    visited = {}
    for (v,t) ∈ sources(G_s) do                ▷ sources are nodes with no incoming edges
        GENERAL_REC((v,t), cum, G, G_s, del, sig, visited)
    end for
    return cum
end function
function GENERAL_REC((v,t), cum, G, G_s, del, sig, visited)
    if (v,t) not in visited then
        su = 0
        for t' ∈ {t''|∃((v,t),(w,t'')) ∈ E_s}, in decreasing order do     ▷ Ordering is necessary only for active walks
            for w ∈ {((v,t),(w,t')) ∈ E_s} do
                GENERAL_REC((w,t'), cum, G, G_s, del, sig, visited)
                su = su + sig[(v,t)] / sig[(w,t')] cum(w,t')
            end for
            INTER_CONTRIBUTION((v,t), t', G_s, cum, del, su, visited)     ▷ for passive walks ignore this instruction
        end for
        cum[(v,t)] = su + del[(v,t)]
        visited.ADD((v,t))
    end if
end function
function INTER_CONTRIBUTION((v,t), t', G_s, cum, del, su, visited)
    t'' = t' − 1
    while t'' ≥ t and bef_{G_s}(v,t'') = t and (v,t'') ∉ visited do
        cum[(v,t'')] = del[(v,t'')] + su
        visited.ADD((v,t''))
        t'' = t'' − 1
    end while
end function
```

Passive shortest foremost and strict variants

For passive shortest foremost variant. We can define $c_s^{fm,pas}(v,t)$ as before and hence compute the values of Eq. (3) as follows:

$$c_s^{fm,pas}(v,t) = \min_{W \in W_{s(v,t)}^{\infty,pas}} (arr(W) \cdot n + len(W)), \quad c_s^{fm}(v) = \min_{t \in [T]} (c_s^{fm,pas}(v,t)).$$

If the set of $s-v$ walks is empty then $c_s^{fm,pas}(v,t) = \infty$. In this way the values of $c_s^{fm,pas}(v)$ as defined in Eq. (3) coincide with $\min_{t \in [T]} c_s^{fm,pas}(v,t)$. We notice that the predecessor graph G from node s in the passive case of sh-∞ is the same as the predecessor graph G' for shortest foremost cost in the passive case. This can be seen by taking any shortest $s-(v,t)$ walk W. then $c^{sh}(W) = c_s^{\infty,pas}(v,t) = len(W)$. Take any $s(v, t)$ walk W'. The last transition of W is of the form (i, v, t) for some $i \in V$. Then $c^{fm}(W') = t \cdot n + len(W') \geq t \cdot n + len(W) = c^{fm}(W)$ thus W is also an optimal $s-(v,t)$ walk that is $c^{fm}(W) = c_s^{fm,pas}(v,t)$. Therefore the set of optimal $s-(v,t)$ walks in terms of $c_s^{fm,pas}(v,t)$ coincides with the one of $c_s^{\infty,pas}(v,t)$. However, the set of optimal $c_s^{fm}(v)$ will be differ from the set of $c_s^{sh}(v)$. Then, the same results and proofs then hold in the same way except the values of $B(v, t)$ become different. For strict version of all variants considered. The only change to be made in Algorithm 1 is on Line 7 by replacing $t' \geq t$ with $t' > t$. The extension of the recurrence in Proposition 3 is immediate as it is the case in Rymar et al. (2021).

The main reason why our formalism does not hold on the active variant of shortest foremost walks is the fact that the recurrence to count to compute optimal $s-(v,t)$ does not hold. This is due to the lack of prefix-optimality. That is the prefix of an optimal $s-(v,t)$ walk is not necessarily an optimal walk. For instance, on the graph of Fig. 1, $c_a(d,6) = 30 + 3 = 33$ and the walk $W = a \xrightarrow{1} b \xrightarrow{5} c \xrightarrow{6} d$ is an optimal $a-(d,6)$ walk (i.e $W \in \mathcal{W}_{a(d,6)}$), since $arr(W) \cdot n + len(W) = 33$. Now, removing the last edge of W yields $W' = a \xrightarrow{1} b \xrightarrow{5} c$ but W' is not an optimal $a-(c,5)$ walk (i.e $W' \notin \mathcal{W}_{a(c,5)}$) since $arr(W') \cdot n + len(W') = 27$. In fact, the optimal cost $c_a(c,5) = 10 + 2 = 12$ using the walk $W'' = a \xrightarrow{1} b \xrightarrow{2} c$.

Conclusion

Our results leave an open question on whether it is possible to characterize cost functions that can be solved using a temporal BFS as we did for the three variants of this paper in the same vein of Rymar et al. (2021). In conclusion, theoretically our work sheds some light on the algorithmic complexity to compute the betweenness centrality of all temporal nodes in a graph and it seems that one can hardly compute this measure on very large graphs due to its theoretic barriers. On a practical level, this work shows how introducing active variants of classical temporal betweenness centrality definitions gives different results on realworld datasets and opens new algorithmic perspectives.

Therefore, it is appealing to approximate this measure for large graphs some work has been done recently in this direction in Santoro and Sarpe (2022) but there is more that could be done in this direction.

Our results leave an open question on whether it is possible to characterize cost functions that can be solved using a temporal BFS as we did for the three variants of this paper in the same vein of Rymar et al. (2021).

Our results improve the theoretical time analysis of previously known methods to compute the temporal betweenness centrality on shortest paths variants. It would be interesting to know if these results could be improved or if it is not the case to extend known hardness complexity results on static betweenness centrality such that Borassi et al. (2016) to the temporal case.

Another direction is to look for guaranteed approximations to the temporal betweenness centrality which started to be studied recently in Santoro and Sarpe (2022) and Cruciani (2023).

Finally, the temporal betweenness centrality has been defined in different time dependent formalisms such that Stream Graphs Latapy et al. (2018) and Simard et al. (2021) that allow for continuous time and it would be interesting to find out if the same kind of results hold in that setting as well.

Acknowledgements
The author would like to thank Matthieu Latapy and Clémence Magnien for their support and the many discussions we had around this subject. The author is also grateful to Maciej Rymar for taking time to explain their results on this subject and kind conversations. The author would like to thank anonymous reviewers for their comments that helped clarify ideas in the paper and improve its general shape.

Author contributions
The only author contributed entirely to the manuscript.

Declarations

Competing interests
The authors declare no competing interests.

Received: 6 October 2023 Accepted: 25 November 2024
Published online: 15 April 2025

References

Borassi M, Crescenzi P, Habib M (2016) Into the square: on the complexity of some quadratic-time solvable problems. Electron Notes Theoret Comput Sci 322:51–67
Brandes U (2001) A faster algorithm for betweenness centrality. J Math Sociol 25(2):163–177
Burt RS (2004) From structural holes: the social structure of competition. New Econ Sociol Reader, pp 325–348
Buß S, Molter H, Niedermeier R et al (2020) Algorithmic aspects of temporal betweenness. In: Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining, pp 2084–2092
Casteigts A, Himmel AS, Molter H et al (2021) Finding temporal paths under waiting time constraints. Algorithmica 83(9):2754–2802
Cruciani A (2023) On approximating the temporal betweenness centrality through sampling. arXiv preprint arXiv:2304.08356
Freeman LC (1977) A set of measures of centrality based on betweenness. Sociometry, pp 35–41
Girvan M, Newman ME (2002) Community structure in social and biological networks. Proc Natl Acad Sci 99(12):7821–7826
Kim H, Anderson R (2012) Temporal node centrality in complex networks. Phys Rev E 85(2):026107
Kostakos V (2009) Temporal graphs. Phys A 388(6):1007–1023
Kujala R, Weckström C, Darst RK et al (2018) A collection of public transport network data sets for 25 cities. Sci Data 5(1):1–14
Latapy M, Viard T, Magnien C (2018) Stream graphs and link streams for the modeling of interactions over time. Soc Netw Anal Min 8(1):1–29
Leydesdorff L (2007) Betweenness centrality as an indicator of the interdisciplinarity of scientific journals. J Am Soc Inform Sci Technol 58(9):1303–1319
Magnien C, Tarissan F (2015) Time evolution of the importance of nodes in dynamic networks. In: Proceedings of the 2015 IEEE/ACM international conference on advances in social networks analysis and mining 2015, pp 1200–1207
Narayanan S (2005) The betweenness centrality of biological networks. PhD thesis, Virginia Tech
Puzis R, Altshuler Y, Elovici Y et al (2013) Augmented betweenness centrality for environmentally aware traffic monitoring in transportation networks. J Intell Transp Syst 17(1):91–105
Rymar M, Molter H, Nichterlein A et al (2021) Towards classifying the polynomial-time solvability of temporal betweenness centrality. In: international workshop on graph-theoretic concepts in computer science, Springer, pp 219–231
Santoro D, Sarpe I (2022) Onbra: rigorous estimation of the temporal betweenness centrality in temporal networks. Proc ACM Web Conf 2022:1579–1588

Simard F, Magnien C, Latapy M (2021) Computing betweenness centrality in link streams. arXiv preprint arXiv:2102.06543
Tang J, Musolesi M, Mascolo C et al (2010) Analysing information flows and key mediators through temporal centrality metrics. In: Proceedings of the 3rd workshop on social network systems, pp 1–6
Tsalouchidou I, Baeza-Yates R, Bonchi F et al (2020) Temporal betweenness centrality in dynamic graphs. Int J Data Sci Anal 9:257–272
Yoon J, Blumer A, Lee K (2006) An algorithm for modularity analysis of directed and weighted biological networks based on edge-betweenness centrality. Bioinformatics 22(24):3106–3108

Publisher's Note