Technische Universität Berlin

# Algorithms and Experiments for Betweeness Centrality in Tree-Like Networks

## Bachelorarbeit

**von Alexander Dittmann**

zur Erlangung des Grades „Bachelor of Science“
(B. Sc.) im Studiengang Informatik

Betreuer:
Matthias Bentert
Dr. André Nichterlein
Prof. Dr. Rolf Niedermeier

Erstgutachter: Prof. Dr. Rolf Niedermeier
Zweitgutachter: Prof. Dr. Toby Walsh

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und eigenhändig sowie ohne unerlaubte fremde Hilfe und ausschließlich unter Verwendung der aufgeführten Quellen und Hilfsmittel angefertigt habe.

Die selbständige und eigenständige Anfertigung versichert an Eides statt:

Berlin, den

______________________________
Unterschrift

## Zusammenfassung

Die *Betweenness Centrality* ist ein Standarmaß bei der Analyse von Netzwerken, um die Wichtigkeit von Knoten in einem Netzwerk einzustufen. Die Betweenness Centrality repräsentiert die relative Anzahl an kürzesten Pfaden, auf dem ein Knoten in einem Netzwerk liegt. Das Ermitteln von Knoten mit hoher Betweenness Centrality, und damit das Finden von Knoten, die wichtig für das Netzwerk sind, hat eine Reihe von Anwendungen in verschiedenen wissenschaftlichen Feldern wie Soziologie, Biologie und Informatik.

Die aktuell schnellsten Algorithmen zum Berechnen der Betweenness Centrality für alle Knoten in einem Netzwerk haben eine Laufzeit von $O(n \cdot m)$, wobei $n$ die Anzahl der Knoten und $m$ die Anzahl der Kanten in dem Netzwerk ist. Diese Laufzeit ist jedoch nicht praktikabel, wenn sehr große Instanzen von Netzwerken analysiert werden müssen. Dies stellt eine Motivation dar, um schnellere Algorithmen zu finden oder vorhandene zu verbessern, sodass die Betweenness Centrality auch in großen Instanzen in angemessener Zeit berechnet werden kann.

Ein solcher $O(n \cdot m)$-Algorithmus ist der von Brandes [J Math Sociol 2001]. Baglioni u. a. [ASONAM 2012] verbesserten diesen, indem sie Grad-Eins-Knoten aus dem Netzwerk löschen, bevor sie Brandes Algorithmus auf dem Netzwerk ausführen. Falls genug Knoten gelöscht werden können, kann das eine wesentliche Beschleunigung darstellen. In dieser Arbeit möchten wir diese Idee weiter ausführen. Wir stellen einen parametrisierten Algorithmus in Hinsicht auf die *Feedback Edge Number* vor, der das Löschen von einigen Knoten von Grad zwei miteinbezieht.

## Abstract

When it comes to analyzing networks the *betweenness centrality* is a standard measure to rank the importance of the vertices in a network. The betweenness centrality represents the relative number of shortest paths a vertex lies on in a network. Finding vertices with a high betweenness centrality and thus vertices that are important for the network, has several applications in different fields of science, for example in sociology, biology and computer science.

The currently best known algorithms for computing the betweenness centrality for each vertex of a network have a running time of $O(n \cdot m)$, where $n$ is the number of vertices and $m$ is the number of edges in the network. This running time however is not feasible for very big instances of networks. This motivates to find faster algorithms or to improve existing algorithms to make the betweenness centrality a usable method of analysis even for these instances which are currently to big for practical use.

One such $O(n \cdot m)$-algorithm is the one of Brandes' [J Math Sociol 2001]. Baglioni et al. [ASONAM 2012] improved Brandes' algorithm by deleting degree-one vertices in the network before running Brandes' algorithm on the network. If enough vertices are deleted, then this can be a significant speed up. In this work we follow this idea. In particular we provide a parameterized algorithm with respect to the *feedback edge number*, which involves the deletion of some vertices of degree two.

# Contents

# 1 Introduction

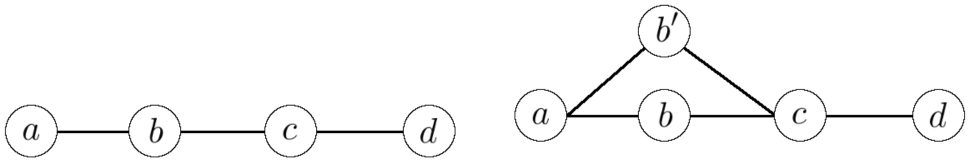
Networks, for example social networks, are often represented as graphs, where the vertices of the graph represent the entities in the network. If there is an edge between two vertices, then the respective entities are in some kind of relationship. This relationship can represent any circumstances: In a social network like Facebook, two people, the entities, could be connected with an edge, if they were “friends”. Methods of graph theory can be applied to the graphs of the networks to analyze them and get various information about the entities or their relationships. One of such information are the so called *centrality indices*, the *betweenness centrality* being one of them. For an overview of the other indices, see [Bra01] for example. Centrality indices were already mentioned by Bavelas [Bav48] in 1948 but without providing a formal definition of them. This was done by Freeman [Fre77] in 1977 for example. Freeman also provided a formal definition of the betweenness centrality. The betweenness centrality is a measure, of how central a vertex is in the graph. This measure is calculated by, roughly said, counting the number of shortest paths that the vertex lies on. For a graph $G = (V, E)$, the exact value of the betweenness centrality $C_B(v)$ for a given vertex $v \in V$ is obtained by adding up for each pair $(s, t)$, $s \neq v \neq t \in V$, of vertices the ratio of the number of shortest paths between $s$ and $t$ that $v$ lies on and the total amount of shortest paths between $s$ and $t$. The formal definition is as follows:

$$C_B(v) = \sum_{\substack{s,t \in V \\ s \neq v \neq t}} \frac{\sigma_{st}(v)}{\sigma_{st}}, \tag{1}$$

where $\sigma_{st}$ denotes the number of shortest path between $s$ and $t$ and $\sigma_{st}(v)$ denotes the number of shortest paths between $s$ and $t$ where $v$ lies on. A formalization of the problem we deal with in this work is:

Betweenness Centrality
**Input:** An undirected graph $G = (V, E)$
**Task:** For each $v \in V$ calculate the betweenness centrality $C_B(v) = \sum_{\substack{s,t \in V \\ s \neq v \neq t}} \frac{\sigma_{st}(v)}{\sigma_{st}}$.

The higher the betweenness centrality of a vertex is in a network, the more shortest paths the vertex lies on. The betweenness centrality is motivated by the assumption that information flow in a network follows shortest paths [Bag+12]. Hence, if a vertex lies on many shortest paths, it has a high potential to influence or control the information flow in the network. For a short example, see Figure 1. In Figure 1a the vertex $b$ has a betweenness centrality of four, since $b$ lies on all shortest paths between four pairs of vertices, which are $(a, c)$, $(a, d)$, $(c, a)$ and $(d, a)$ (each pair is counted twice). In Figure 1b, a new vertex $b'$ is added. Now, between each of the just mentioned pairs of vertices, there are two shortest paths, and $b$ only lies on one of them. That is, why its betweenness centrality reduces to two: Both $b$ and $b'$ now share the connections between $a$ and $c$ and $a$ and $d$; $b$ does not have exclusive influence on these connections anymore.




(a) The betweenness centrality of $b$ is four

(b) The betweenness centrality of $b$ is only two

Figure 1: Example for betweenness centrality in two similar graphs

Betweenness centrality plays an important role in different fields of science. In brain networks, it could be used to find neural links with a high information flow. These links may represent important anatomical or functional connections between different regions in the brain [RS10]. In social or biological networks in general, betweenness measures can be used for finding community structures [For10; NG04]. Community structures are clusters of vertices with a high amount of edges between the vertices of a cluster, while between the clusters, there are rather few links. Those links are important for the information flow between the groups and hence, the vertices being part of these links are more likely to have a high betweenness centrality. Since these networks are often big, a fast computation of betweenness centrality is desirable.

If we wanted to compute the betweenness centrality for all vertices of a graph and look at formula (1), a naive approach of doing so would take $O(n^3)$ time in total. This is only feasible for relatively small networks. If we look at bigger networks though, with (hundreds of) thousands of vertices, this running time is not feasible for practical use any more. In 2001 Brandes [Bra01] introduced an algorithm for computing the betweenness centrality of each vertex in a graph in $O(n \cdot m)$ time, where $n$ denotes the number of vertices and $m$ denotes the number of edges in the graph. Since (social) networks are rather sparse [Bra01], this is a significant improvement to the hitherto known naive approaches.

In 2012, Baglioni et al. [Bag+12] introduced an improvement for Brandes' algorithm. Before running the algorithm of Brandes on the graph, they recursively delete all vertices of degree one, leaving a graph with only vertices of degree at least two. This reduces the number of vertices on which Brandes' algorithm has to be performed and thus improves the running time of the computation. The performance improvement directly depends on the number of degree-one vertices in the graph.

Now, our approach is to take the next step and continue the work of Baglioni et al. [Bag+12]. We want to delete as many vertices of degree two as possible in the input graph to shrink the graph even more and thus, to reduce the number of vertices that Brandes' algorithm needs to be run on. We additionally give a theoretical analysis by using the *feedback edge set* of the graph. This is a set of edges such that removing it from the graph results in an acyclic graph. The size $k$ of the smallest such set is called the *feedback edge number*. In this work, we want to provide an algorithm with a running time of the form $O(k^{O(1)} \cdot n^2)$. This follows the idea of "FPT in P": For a problem solvable in polynomial time $O(n^c)$, we want to find a parameter $k$ and an algorithm, such that the problem is solvable in $O(f(k) \cdot n^{c'})$, where $c' < c$ and $f$ only being dependent from $k$. If $k$ is small enough, then this is an improvement to the running time of $O(n^c)$. Our parameter $k$ is the feedback edge number of the input graph and the algorithm that we present in this work has a running time of $O(k^2 \cdot n^2)$. For a more comprehensive explanation of "FPT in P", see Giannopoulou et al. [Gia+17]. At this point, we want to formalize the theorem that we proof with this work:

**Theorem 1.1.** *The betweenness centrality for all vertices in a graph can be computed in $O(k^2 \cdot n^2)$ time, where $k$ is the feedback edge number of the graph.*

The rest of this work is structured as follows: In Section 2 we list formulas and variables that we use in this work. In Section 3 we give a an overview of our algorithm. We give definitions for *balloons* and *chains* as well as a brief pseudo code to put all of our steps in a defined order. In Section 4 we show how we handle the just mentioned balloons. In Section 5 we do the same for chains. Section 5 will be the most extensive one in this work. In Section 6 we will give a conclusion and ideas on how to further improve our algorithm.

## 2 Preliminaries

Here, we define the formulas and variables that we use in this work.

An undirected graph is a pair $G = (V, E)$. In this context, we denote by

$V$ the *vertex set* of $G$;

$E$ the *edge set* of $G$ with $E \subseteq \binom{V}{2}$; for an edge $e = \{u, v\} \in E$ the two vertices $u$ and $v$ are called *endpoints* of $e$;

$n_G$ the number $|V|$ of *vertices*;

$m_G$ the number $|E|$ of *edges*;

$\deg_G(v)$ the *degree* of $v$ in $G$;

$\mathrm{V}^{=1}$ the set of all vertices in $V$ with degree one;

$\mathrm{V}^{=2}$ the set of all vertices in $V$ with degree two;

$\mathrm{V}^{\geq 3}$ the set of all vertices in $V$ with degree at least three;

$\sigma_{vw}$ the number of shortest paths between $v$ and $w$;

$\sigma_{vw}(u)$ the number of shortest paths between $v$ and $w$ that include vertex $u$; for convenience we set $\sigma_{vw}(v) = \sigma_{vw}(w) = \sigma_{vw}$;

$d_G(v, w)$ the *distance* of $v$ and $w$, i. e. the length of a shortest path between $v$ and $w$;

$S(T)$ the set of all elements that appear in the tuple $T$, i. e. for $T = (t_1, ..., t_\ell)$ we have $S(T) = \{t_1, ..., t_\ell\}$ ;

$S^*(T)$ the set of all elements that appear in the tuple $T$ except the first and the last, i. e. for $T = (t_1, ..., t_\ell)$ we have $S^*(T) = \{t_2, ..., t_{\ell-1}\}$ ;

$G - \{v\}$ the induced subgraph of $G$, obtained by removing $v \in V$ and all its edges from $G$;

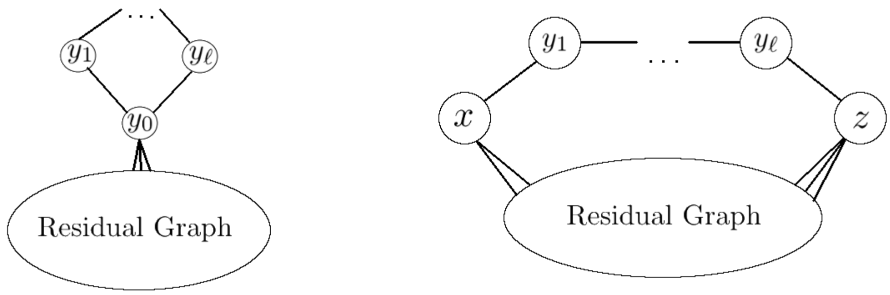
$G[V']$ the induced subgraph of $G$, obtained by removing all vertices $v \in V$, $v \notin V' \subseteq V$ and all its edges from $G$;

If the graph $G$ is clear from the context, then we will omit the subscript $G$.

## 3 Overview of the algorithm

As mentioned in the introduction, Baglioni et al. [Bag+12] improved Brandes' algorithm [Bra01] by recursively deleting all degree-one vertices in the input graph before running Brandes' algorithm on it. For the deletion process they introduce a labeling $p$, $V \to N$, where initially $p(v) = 0$ for all $v \in V$. After deleting a degree-one vertex $v$ in the graph, they increase $p(v')$ by $p(v) + 1$, with $v'$ being the unique neighbour of $v$. Thus, $p(v)$ denotes the number of vertices that were already deleted from the graph and directly connected to $v$. This labeling is needed for calculating the correct betweenness centrality for the remaining vertices later, when running Brandes' algorithm on the graph. Baglioni et al. slightly modify Brandes' algorithm to involve the labeling $p$. Besides, in the deletion process, they already compute the final betweenness centrality of each deleted vertex. See Baglioni et al. [Bag+12] for a more detailed description. The main ideas behind that procedure are that they shrink the input graph before running Brandes algorithm on it by deleting vertices of degree one, that the shrinking process can be done in almost linear time and that the deletion of those vertices does not alter the betweenness centrality of the remaining vertices in the graph.

In this work, we want to further improve the algorithm of Baglioni et al. [Bag+12]. Our algorithm starts with the same procedure as Baglioni's algorithm does: We delete all vertices of degree one in the graph and apply the labeling $p$ on all vertices left, with a small difference: We want $p(v)$ to represent the number of vertices that were directly connected to $v$ (as in Baglioni's algorithm) plus $v$ itself. Therefore, after removing the degree-one vertices, we increase each value for $p$ by one. After the deletion of all degree-one vertices, the input graph contains only vertices with a degree of at least two. Our goal is to delete as many degree-two vertices as possible to further reduce the number of vertices that Brandes' algorithm needs to be run on. For this purpose, we distinguish between two types of subgraphs in which degree-two vertices may appear:




(a) a *balloon*

(b) a *chain*

Figure 2: A balloon and a chain

**Definition 3.1.** A *balloon* is a path $(y_0, y_1, ..., y_\ell, y_0)$ where $y_0 \in V$ and $y_1, ..., y_\ell \in \mathrm{V}^{=2}$.

**Definition 3.2.** A *chain* is a path $(x, y_1, ..., y_\ell, z)$ where $x, z \in \mathrm{V}^{\geq 3}$ and $y_1, ..., y_\ell \in \mathrm{V}^{=2}$ and $x \neq z$.

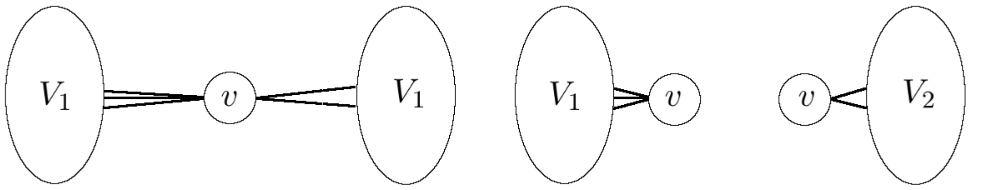
See Figure 2a and Figure 2b for a visualization of a balloon and a chain, respectively. These subgraphs can be found in linear time in the input graph, as we will see later. In addition, every vertex of degree two is either part of a balloon or part of a chain. If there are only vertices of degree two left in the graph, then the whole graph is a balloon. At this point we give a more detailed look on the feedback edge set of the graph. With its help we can bound the number of chains and the number of vertices with degree at least three in the graph.

**Feedback Edge Set and Feedback Edge Number** As explained in Section 1, the feedback edge set of a graph is a set of edges that needs to be removed to make the graph acyclic. The feedback edge number is the size of a smallest such set. Now, consider the residual graph after we have removed all vertices of degree one from it using the procedure of Baglioni et al. [Bag+12]. The vertices in the residual graph all have a degree of at least two. If follows from the work of Mertzios et al. [Mer+17, proof of Thm 2.3] that the number of chains in the residual graph is linear in the feedback edge number $k$, i. e. $|C| \in O(k)$. The same holds for the vertices of degree at least three, i. e. $|\mathrm{V}^{\geq 3}| \in O(k)$, too.

In the next paragraph we show how we can remove balloons from the graph. Chains are handled later in Section 5, because they cannot be simply deleted from the graph.

### 3.1 Splitting balloons from the graph

The behavior of balloons in the graph is similar to that of degree-one vertices. This allows us to remove them for similar reasons as degree-one vertices can be deleted from $G$. Vertices of degree one can be deleted from $G$, because all shortest paths from a vertex of degree one pass through its unique neighbour. For balloons, this is similar: All shortest




$G = (V_1 \cup V_2, E)$ $\quad$ $G_1 = (V_1, E_1)$ $\quad$ $G_2 = (V_2, E_2)$

(a) The graph before splitting $\quad$ (b) The two new graphs after splitting

Figure 3: Splitting a graph into two

paths from each vertex $y_1, ..., y_\ell$ in the balloon to each vertex outside that balloon pass through $y_0$. After removing a balloon from the graph we just have to increase $y_0$'s value for $p$ by the size of the balloon (and all vertices that were originally connected to the balloon). We do this step iteratively, since after "popping" a balloon, that is deleting all vertices except $y_0$ from the input graph, there could be a new one. At this point, we want to introduce the following helpful lemma. See Figure 3 for a visualization of it. In Figure 3a vertex $v$ is the only connection between the vertices in $V_1$ and $V_2$. All shortest paths between vertices of $V_1$ and vertices of $V_2$ pass through $v$. If we split the graph into two as shown in Figure 3b we have to make sure that after performing Brandes' algorithm on the graphs the betweenness centrality for the vertices in them are not altered. Therefore, we have to increase the betweenness centrality of $v$ by the product of the number of vertices in $V_1$ and $V_2$. Then, because we delete a number of vertices from the graph, we yet have to increase the value of $p(v)$ by the sum of all $p$-values of all vertices in $V_1$ or $V_2$, respectively. This is expressed in Lemma 3.3:

**Lemma 3.3** (Split Lemma)**.** *Let $G = (V, E)$ be a connected graph and let $v \in V$ be any vertex, such that the graph $G - v$ is disconnected. Let $G' = (V', E')$ be one connected component of $G - v$. Set $G_1 = (V_1, E_1) := G[V' \cup \{v\}]$ and $G_2 = (V_2, E_2) := G[V \setminus V']$. Also, let $p_{G_1}(v) = \sum_{w \in V_2} p_G(w)$ and let $p_{G_2}(v) = \sum_{w \in V_1} p_G(w)$. For all $u_1 \in V_1 \setminus \{v\}$ and $u_2 \in V_2 \setminus \{v\}$, let $p_{G_1}(u_1) = p_G(u_1)$ and $p_{G_2}(u_2) = p_G(u_2)$.*

*Then, for $u \in V$ we have:*

$$C_B^G(u) = \begin{cases} C_B^{G_1}(u) + C_B^{G_2}(u) + 2 \sum_{w \in V_1 \setminus \{v\}} p(w) \sum_{w \in V_2 \setminus \{v\}} p(w), \text{ if } u = v \\ C_B^{G_1}(u), \text{ if } u \in V_1 \text{ and } u \neq v \\ C_B^{G_2}(u), \text{ if } u \in V_2 \text{ and } u \neq v \end{cases}$$

*Proof.* First, we will show that $C_B^G(u) = C_B^{G_1}(u)$ for each $u \in V$, $u \neq v$. The proof of $C_B^G(u) = C_B^{G_2}(u)$ is analogous. Then, we will show the correctness of $C_B^G(v) = C_B^{G_1}(v) + C_B^{G_2}(v) + 2 \sum_{w \in V_1 \setminus \{v\}} p(w) \sum_{w \in V_2 \setminus \{v\}} p(w)$.

Let $u \in V_1 \setminus \{v\}$:

$$
\begin{aligned}
C_B^{G_1}(u) &= \sum_{\substack{s,t \in V_1 \\ s \neq u \neq t}} p_{G_1}(s) p_{G_1}(t) \frac{\sigma_{st}(u)}{\sigma_{st}} \\
&= \sum_{\substack{s,t \in V_1 \setminus \{v\} \\ s \neq u \neq t}} p_{G_1}(s) p_{G_1}(t) \frac{\sigma_{st}(u)}{\sigma_{st}} + 2 \sum_{\substack{s \in V_1 \setminus \{v\} \\ s \neq u \neq v}} p_{G_1}(s) p_{G_1}(v) \frac{\sigma_{sv}(u)}{\sigma_{sv}} \\
&\overset{\text{(i)}}{=} \sum_{\substack{s,t \in V_1 \setminus \{v\} \\ s \neq u \neq t}} p_G(s) p_G(t) \frac{\sigma_{st}(u)}{\sigma_{st}} + 2 \sum_{\substack{s \in V_1 \setminus \{v\} \\ s \neq u \neq v}} p_G(s) p_{G_1}(v) \frac{\sigma_{sv}(u)}{\sigma_{sv}} \\
&\overset{\text{(ii)}}{=} \sum_{\substack{s,t \in V_1 \setminus \{v\} \\ s \neq u \neq t}} p_G(s) p_G(t) \frac{\sigma_{st}(u)}{\sigma_{st}} + 2 \sum_{\substack{s \in V_1 \setminus \{v\} \\ s \neq u \neq v}} p_G(s) \Big( \sum_{w \in V_2} p_G(w) \Big) \frac{\sigma_{sv}(u)}{\sigma_{sv}} \\
&\overset{\text{(iii)}}{=} \sum_{\substack{s,t \in V_1 \setminus \{v\} \\ s \neq u \neq t}} p_G(s) p_G(t) \frac{\sigma_{st}(u)}{\sigma_{st}} + 2 \sum_{\substack{s \in V_1 \setminus \{v\} \\ w \in V_2 \\ s \neq u \neq v}} p_G(s) p_G(w) \frac{\sigma_{sw}(u)}{\sigma_{sw}} \\
&\overset{\text{(iv)}}{=} \sum_{\substack{s,t \in V_1 \setminus \{v\} \\ s \neq u \neq t}} p_G(s) p_G(t) \frac{\sigma_{st}(u)}{\sigma_{st}} + 2 \sum_{\substack{s \in V_1 \setminus \{v\} \\ w \in V_2 \\ s \neq u \neq v}} p_G(s) p_G(w) \frac{\sigma_{sw}(u)}{\sigma_{sw}} \\
&\quad + \sum_{\substack{s,t \in V_2 \setminus \{v\} \\ s \neq u \neq t}} p_G(s) p_G(t) \frac{\sigma_{st}(u)}{\sigma_{st}} \\
&= \sum_{\substack{s,t \in V \\ s \neq u \neq t}} p_G(s) p_G(t) \frac{\sigma_{st}(u)}{\sigma_{st}} = C_B^G(u)
\end{aligned}
$$

(i): $p_{G_1}(w_1) = p_G(w_1)$ for each $w_1 \in V_1 \setminus \{v\}$ and $p_{G_2}(w_2) = p_G(w_2)$ for each $w_2 \in V_2 \setminus \{v\}$

(ii): $p_{G_1}(v) = \sum_{w \in V_2} p_G(w)$

(iii): Each shortest path from $w \in V_2$ to $s, t \in V_1 \setminus \{v\}$ has to pass through $v$. Thus, if $u$ lies on a shortest path from $v$ to $s$ or $t$, then $u$ will also lie on the shortest path from $w$ to $s$ or $t$. So, $\frac{\sigma_{vt}(u)}{\sigma_{vt}} = \frac{\sigma_{wt}(u)}{\sigma_{wt}}$ and $\frac{\sigma_{sv}(u)}{\sigma_{sv}} = \frac{\sigma_{sw}(u)}{\sigma_{sw}}$.

(iv): If $s, t \in V_2 \setminus \{v\}$ and $u \in V_1 \setminus \{v\}$, then $\sigma_{st}(u) = 0$, because each shortest path from $s$ to $t$ "stays" in $V_2$: There is only one vertex connecting the vertices in $V_1$ and $V_2$ and a shortest path never covers the same vertex twice.

For $v$:

$$\begin{aligned}
&C_B^{G_1}(v) + C_B^{G_2}(v) + 2\left(\sum_{w \in V_1 \setminus \{v\}} p_G(w)\right)\left(\sum_{w \in V_2 \setminus \{v\}} p_G(w)\right) \\
=&C_B^{G_1}(v) + C_B^{G_2}(v) + 2 \sum_{\substack{w_1 \in V_1 \setminus \{v\} \\ w_2 \in V_2 \setminus \{v\}}} p_G(w_1)p_G(w_2) \\
\overset{(v)}{=}&C_B^{G_1}(v) + C_B^{G_2}(v) + 2 \sum_{\substack{w_1 \in V_1 \setminus \{v\} \\ w_2 \in V_2 \setminus \{v\}}} p_G(w_1)p_G(w_2)\frac{\sigma_{w_1 w_2}(v)}{\sigma_{w_1 w_2}} \\
=&\sum_{\substack{s,t \in V_1 \\ s \neq v \neq t}} p_{G_1}(s)p_{G_1}(t)\frac{\sigma_{st}(v)}{\sigma_{st}} + \sum_{\substack{s,t \in V_2 \\ s \neq v \neq t}} p_{G_2}(s)p_{G_2}(t)\frac{\sigma_{st}(v)}{\sigma_{st}} \\
&+ 2 \sum_{\substack{w_1 \in V_1 \\ w_2 \in V_2 \\ w_1 \neq v \neq w_2}} p_G(w_1)p_G(w_2)\frac{\sigma_{w_1 w_2}(v)}{\sigma_{w_1 w_2}} \\
=&\sum_{\substack{s,t \in V \\ s \neq v \neq t}} p_G(s)p_G(t)\frac{\sigma_{st}(v)}{\sigma_{st}} = C_B^G(v)
\end{aligned}$$

(v): Vertex $v$ is part of each shortest path from $w_1 \in V_1 \setminus \{v\}$ to $w_2 \in V_2 \setminus \{v\}$. Thus, $\frac{\sigma_{w_1 w_2}(v)}{\sigma_{w_1 w_2}} = 1$.

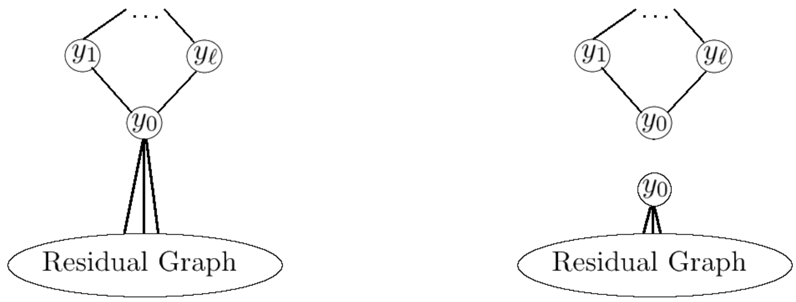
□

Notice that for a balloon $B = (y_0, y_1, ..., y_\ell, y_0)$ vertex $y_0$ is a separator as described in Lemma 3.3. Basically, Lemma 3.3 generalizes the procedure of Baglioni et al. [Bag+12] treating vertices of degree one: It removes vertices from the graph (and does some extra computations) without altering the final betweenness centrality of the remaining vertices. We will use Lemma 3.3 to separate the balloons from the graph without influencing the betweenness centrality for the residual graph.

Before we can actually begin with removing balloons from the graph we have to find them first. In the next paragraph we show how to do so. Note that we search for all chains in the graph parallel, since we of course need to find the chains, too, to process them. We also mention that in this work we do not remove the chains from the graph and hence, we do not remove all vertices of degree two from the graph. In Section 5 we give a detailed explanation on how we deal with chains.

## 3.2 Finding balloons and chains

Before continuing with Section 4 and Section 5, we show how to find all balloons and chains in a graph $G$.




(a) The graph before the removal of $B$ (b) The graph after the removal of $B$

Figure 4: Removing a balloon

Consider any vertex $v \in V$. If $\deg_G(v) = 2$, then iteratively look at the left and right neighbour of $v$, until there are neighbours $v_l$ and $v_r$ with a degree greater than two. If $v_l = v_r$, we found a balloon. Let $B = (y_0, y_1, ..., y_\ell, y_0)$ be the found balloon. If $v_l \neq v_r$, we found a chain. Let $C = (x, y_1, ..., y_\ell, z)$ be the found chain.

If we find a balloon, then we will remove it from $G$ according to Lemma 3.3. See Figure 4 for a visualization. Afterwards, we have to check $y_0$ again, since in the residual graph the degree of $y_0$ is decreased by two. If then $\deg_G(y_0) = 1$, then we will just remove $y_0$, too, in the same way as Baglioni et al. [Bag+12] remove the degree-one vertices from the graph. But if $\deg_G(y_0) = 2$, then we have to do the same procedure as above again. If, additionally, $y_0$ was an endpoint of a chain, then we have to extend that chain instead of creating a new one. Let $\mathcal{B}$ be the set of all balloons and let $\mathcal{C}$ the set of all chains. If we keep a mapping of which vertex is part of which chain, then this part can be done in $O(n + m)$ time.

To complete this section we give a short summary of the next steps followed by a brief pseudo code in Section 3.3. The pseudo code will help later, when linking the sections of this work with their respective part of the pseudo code:

- In Section 4 we will show how we process the found balloons.
- Section 4 is a preparation to the more extensive part of dealing with the chains in Section 5: After deleting all balloons the remaining vertices of degree two are all part of either chain. Then, we are ready to run the modified version of Brandes' algorithm on the shrinked graph. At last, we will process the chains. In Section 5.5 we provide the overall running time of our algorithm.
- In Section 6 we give a conclusion followed by some ideas on how to further improve the algorithm.

## 3.3 Outline of the Algorithm

We provide a brief a pseudo code in Algorithm 1. The pseudo code does not cover every detail of our computations. By looking at the code, the reader can quickly get an overview of what we do in this work. This is suggested before continuing with reading the following sections. In the first three lines we do the preparation work: Initializing the needed variables and removing all vertices of degree one from the graph. In lines 5-13 we search for balloons and chains in the graph. If we find a balloon, then we remove it from the graph according to Lemma 3.3 and add the balloon to the set of all balloons. If we find a chain, then we just add it to the set of all chains. In lines 15-17 we compute the betweenness centrality of the vertices in the balloons. In the last three loops we deal with the found chains (see Section 5).

## 4 Popping balloons: Dealing with cyclic structures

In Section 3 we already showed how to find balloons and how to remove them from the graph using Lemma 3.3. In this section we calculate the appropriate values for the betweenness centrality for the vertices in the balloons. Therefore, look at any $B = (y_0, y_1, ..., y_\ell, y_0) \in \mathcal{B}$. Since we removed the balloon from the graph, we can look at the balloon isolated from the rest of the graph. The respective line in Algorithm 1 of this part is line 16.

**Handling the balloon** Before starting with the computations of the betweenness centrality for the vertices in $B$, we construct two tables $\mathrm{t}_{\mathrm{left}}$ and $\mathrm{t}_{\mathrm{right}}$ for $B$. They are defined as follows:

$$\mathrm{t}_{\mathrm{left}}(y_i) := \sum_{k=0}^{i} p(y_k)$$

$$\mathrm{t}_{\mathrm{right}}(y_i) := p(y_0) + \sum_{k=i}^{\ell} p(y_k)$$

Intuitively, $\mathrm{t}_{\mathrm{left}}(y_i)$ represents the number of vertices that were originally connected to $y_0, ...y_i$ plus the number of these vertices itself, while $\mathrm{t}_{\mathrm{right}}(y_i)$ represents the number of vertices that were originally connected to $y_i, ..., y_\ell, y_0$ plus the number of these vertices itself. With a naive approach, the above tables would need quadratic time for their construction. Because we want the computations to work in linear time, here is how the tables can be computed iteratively in linear time:

$$\begin{aligned}
\mathrm{t}_{\mathrm{left}}(y_i) &= \begin{cases} p(y_0), \text{ if } i = 0 \\ \mathrm{t}_{\mathrm{left}}(y_{i-1}) + p(y_i), \text{ otherwise} \end{cases} \\
\mathrm{t}_{\mathrm{right}}(y_i) &= \begin{cases} p(y_0), \text{ if } i = 0 \\ \mathrm{t}_{\mathrm{right}}(y_{i+1 \bmod l+1}) + p(y_i), \text{ otherwise} \end{cases} .
\end{aligned} \tag{2}$$

In the next part, for each $y_i$, we compute $C_B^B(y_i)$, the betweenness centrality of the vertex $y_i$ in the balloon $B$.

We first look at $y_0$ and calculate $C_B^B(y_0)$. Next, for each other $y_i$, we iteratively compute $C_B^B(y_i)$. Doing this iteratively ensures a linear running time for the processing

**Algorithm 1** Betweenness Centrality

**Input:** An undirected, unweighted graph $G = (V, E)$
**Output:** $C_B(v)$ for all $v \in V$

$C_B(v) \leftarrow 0$ for all $v \in V$
run Baglionis' procedure on $G$ // delete all vertices of degree one from $G$
p(v) $\leftarrow$ p(v)+1 for $v \in V$ // increase p-value of each vertex by one

**while** there is a balloon or chain **do**
  **if** balloon **then**
    Let $(y_0, y_1, ..., y_\ell, y_0)$ be the balloon
    $(G_1, G_2) \leftarrow$ split$(G, y_0)$ // split found balloon from the graph
    $C_B^G(y_0) \leftarrow C_B^G(y_0) + 2\sum_{w \in V_1 \setminus \{y_0\}} p(w) \sum_{w \in V_2 \setminus \{y_0\}} p(w)$ // increase betweenness centrality of the separator $y_0$
    add balloon to set of all balloons
    $G \leftarrow G_2$
  **else**
    add chain to set of all chains

**for** balloon in balloons **do**
  compute betweenness for balloon // Section 4
run Brandes' modified algorithm on vertices with degree at least three

**for** chain in chains **do**
  **for** $v$ in $V$ **do**
    **if** $deg_G(v) \geq 3$ **then**
      process chain-vertex pair (chain, $v$) // Section 5.2, (Step 1) (Section 5.2)

**for** chain1 in chains **do**
  **for** chain2 in chains **do**
    process chain pair (chain1, chain2) // Section 5.3, (Step 2)

**for** chain in chains **do**
  process single chain(chain) // Section 5.4, (Step 3)

of each balloon. For the following computations we define

$$t(y_i, y_j) := \begin{cases} p(y_i), \text{ if } i = j \\ \mathrm{t_{left}}(y_j), \text{ if } i < j \text{ and } i = 0 \\ \mathrm{t_{left}}(y_j) - \mathrm{t_{left}}(y_{i-1}), \text{ if } i < j \text{ and } i > 0 \\ \mathrm{t_{left}}(y_j) + \mathrm{t_{right}}(y_i) - p(y_0), \text{ if } i > j \end{cases}.$$

The formula $t(y_i, y_j)$ sums up all values for $p$ from $y_i$ to $y_j$, clockwise ($p(y_i) + ... + p(y_j)$, or $p(y_j) + ... + p(y_\ell) + p(y_0) + ... + p(y_i)$, if $j < i$). In the rest of the section we show the following theorem:

**Theorem 4.1.** *Let $B = (y_0, y_1, ..., y_\ell, y_0)$ be a balloon. The betweenness centrality of all vertices in $B$ can be computed in $O(\ell)$ time.*

Depending of whether $\ell$ is even or odd, we have to do slightly different computations. A formal proof can be found at the end of this section.

**Case 1: $\ell$ is even** We compute the betweenness centrality for $y_0$ by constructing a sum. We will first give the definitions and then an explanation. The betweenness centrality to add to $y_0$ is:

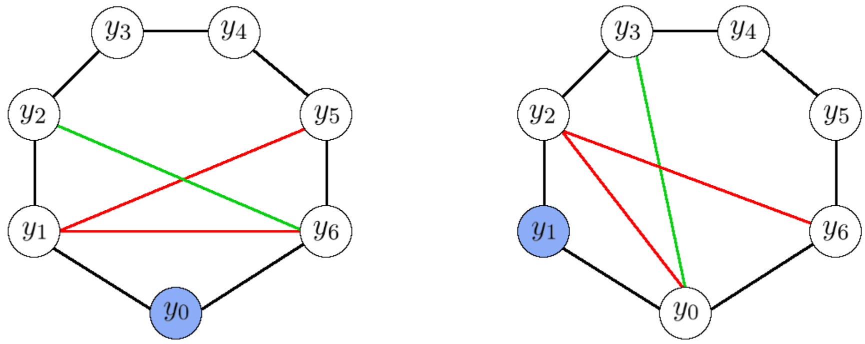
$$C_B^B(y_0) := \sum_{k=1}^{\frac{l}{2}-1} p(y_k) \cdot (\mathrm{t_{right}}(y_{\frac{l}{2}+1+k}) - p(y_0)).$$

From each vertex $y_i$, $1 \le i \le \frac{\ell}{2} - 1$, there is exactly one shortest path to each vertex $y_j$, $\frac{\ell}{2} + 1 + i \le j \le \ell$, that includes $y_0$. Each of these shortest paths has to be considered $p(y_i) \cdot p(y_j)$ times, since we also have to include the shortest paths from the vertices that were originally connected to $y_i$ and $y_j$. If we sum up all possible $p(y_j)$ for a fixed $y_i$, then we get $\mathrm{t_{right}}(\frac{\ell}{2} + 1 + i) - p(y_0)$. The sum above adds up the values for all shortest paths that pass through $y_0$.

For every other $y_i$, $0 < i \le \ell$, we compute the following (starting at $y_1$). Again, the explanation follows:

$$\begin{aligned} C_B^B(y_i) :=& C_B^B(y_{i-1}) \\ & - p(y_i) \cdot t(y_{i-\frac{l}{2} \bmod l+1}, y_{i-2 \bmod l+1}) \\ & + p(y_{i-1}) \cdot t(y_{i+1 \bmod l+1}, y_{i+\frac{l}{2}-1 \bmod l+1}). \end{aligned} \tag{3}$$

Basically, the formula $C_B^B(y_i)$ takes the just computed value for $y_{i-1}$, subtracts the amount of betweenness that is given by all shortest paths that pass through $y_{i-1}$ but do not pass through $y_i$ and adds the amount of betweenness centrality that is given by all shortest paths that do not pass through $y_{i-1}$ but do pass through $y_i$. This process is like shifting a window clockwise, where the window indicates which shortest paths between which pairs of vertices need to be considered for $y_i$. See Figure 5 for a graphical representation of the window. The blue colored vertex is the currently treated vertex.




(a) The window for the here treated vertex $y_0$

(b) The window for the here treated vertex $y_1$

Figure 5: The window indicating the shortest paths that the treated vertex lies on

The red and green lines between two vertices symbolize that between these vertices there is a shortest path where the currently treated vertex lies on. In Figure 5a between $y_1$ and $y_6$ and $y_1$ and $y_5$ there is a shortest path where $y_0$ (the treated vertex) lies on. This is indicated by the two red lines connecting these vertices. Between $y_2$ and $y_6$ there is a shortest path where $y_0$ lies on, too. This is indicated by the green line connecting them. These are all shortest paths that need to be considered for $y_0$. Now, in Figure 5b we treat the next vertex, $y_1$. Therefore, we shift the window by one step clockwise. Afterwards, the red lines now connect $y_2$ with $y_0$ and $y_6$, because between those pairs of vertices there now is a shortest path that $y_1$ (the treated vertex) lies on. The green line connects $y_3$ and $y_0$, because between those vertices there is also a shortest path that includes $y_1$.

**Case 2: $\ell$ is odd** This case is slightly different. In Case 1, there is exactly one shortest path between each pair of vertices. In Case 2, there are pairs of vertices where there are exactly two shortest paths between them. These are the pairs of vertices which are opposite or, expressed mathematically, those $y_i$, $y_j$, $i < j$, where $j - i = \lfloor \frac{\ell}{2} \rfloor$. Each other vertex can only lie on one of these two shortest paths. This is why there is a term with a factor of $\frac{1}{2}$ in the formulas. Again, we first compute the betweenness for $y_0$ by constructing a sum. The sum is:

$$C_B^B(y_0) \coloneqq \sum_{k=1}^{\lfloor \frac{l}{2} \rfloor} p(y_k) \cdot (\mathrm{t}_{\text{right}}(y_{\lceil \frac{l}{2} \rceil + 1 + k \bmod l+1}) - p(y_0)) + \frac{1}{2} p(y_k) \cdot p(y_{\lceil \frac{l}{2} \rceil + k}).$$

Every other $y_i$ is computed recursively as in Case 1. Again, we have pairs of vertices where there are two shortest paths between them. These need to be treated separately, which is why there are again two terms with the factor $\frac{1}{2}$, one of them being subtracted and one being added.

$$
\begin{aligned}
C_B^B(y_i) \coloneqq & C_B^B(y_{i-1}) \\
& - p(y_i)(t(y_{i+\lceil \frac{l}{2} \rceil + 1 \bmod l+1}, y_{i-2 \bmod l+1}) - \frac{1}{2} p(y_{i+\lceil \frac{l}{2} \rceil \bmod l+1})) \\
& + p(y_{i-1})(t(y_{i+1 \bmod l+1}, y_{i+\lfloor \frac{l}{2} \rfloor - 1 \bmod l+1}) + \frac{1}{2} p(y_{i+\lfloor \frac{l}{2} \rfloor \bmod l+1})).
\end{aligned}
$$

Since we only look at each pair of vertices once, all results for $C_B^B(y_i)$, $0 \le i \le \ell$, in this section have to be doubled. Next, we prove the correctness of Theorem 4.1.

*Proof.* We will divide the proof of the formula $C_B^B$ into two parts. In the first part, we show that the value $C_B^B(y_0)$ is correct. This will be our induction basis. In the second step, we show the correctness of any $C_B^B(y_i)$, given that $C_B^B(y_{i-1})$ is already correct. Combining both steps will yield the total correctness of the formula $C_B^B$. Additionally, let $\ell > 2$, because otherwise, the balloon is a clique and the betweenness centrality of each vertex in the balloon is zero. At the end, we proof the linear running time of the computations.

**Step 1: $C_B^B(y_0)$ is correct** First, look at the following observation. The vertex $y_0$ lies on a shortest path between two vertices $y_{i_1}$ and $y_{i_2}$ if the path $y_{i_1}, ..., y_0, y_l, ..., y_{i_2}$ has a maximum length of $\frac{l}{2}$. Otherwise, the path in the opposite direction would be shorter and, thus, would not include $y_0$. Formally, the inequation

$$
i_1 + l - i_2 + 1 \le \frac{l}{2} \tag{4}
$$

has to be true for all pairs $y_{i_1}, y_{i_2}$. The value $i_1$ is the distance from $y_0$ to $y_{i_1}$ (clockwise) while the value $(\ell - i_2 + 1)$ is the distance from $y_0$ to $y_\ell$ (anti-clockwise). Recall, that we used the following formula to compute $C_B(y_0)$:

$$
C_B^B(y_0) = \sum_{k=1}^{\frac{l}{2}-1} p(y_k) \cdot (\mathrm{t}_{\mathrm{right}}(y_{\frac{l}{2}+1+k}) - p(y_0)).
$$

In the formula, $i_1$ corresponds to $k$ and $i_2$ corresponds to $\frac{l}{2} + 1 + k$. So, for these choices of $i_1$ and $i_2$ inequality (4) has to be true:

$$
\begin{aligned}
& & i_1 + l - i_2 + 1 &\le \frac{l}{2} \\
&\iff & k + l - (\frac{l}{2} + 1 + k) + 1 &\le \frac{l}{2} \\
&\iff & k + l - \frac{l}{2} - 1 - k + 1 &\le \frac{l}{2} \\
&\iff & l - \frac{l}{2} &\le \frac{l}{2} \\
&\iff & \frac{l}{2} &\le \frac{l}{2}
\end{aligned} \tag{5}
$$

As we see, the inequality holds.

Since $k$ is bounded by $\frac{l}{2}-1$, $y_{\frac{l}{2}+1+\frac{l}{2}-1} = y_l$ is the last vertex that is included in the formula. For all greater choices of $k$, $y_{\frac{l}{2}+1+k}$ would be left to or equal to $y_0$ and thus, $y_0$ would not be part of the shortest paths from $y_k$ to $y_{\frac{l}{2}+1+k}$. This means that we did not miss any shortest paths in our computations of $C_B^G(y_0)$

**Step 2: $C_B^B(y_i)$ is correct** Consider

$$C_B^B(y_i) \coloneqq C_B^B(y_{i-1})$$
$$-p(y_i)\cdot t(y_{\frac{l}{2}+1+i \bmod l+1}, y_{l+i-1 \bmod l+1})$$
$$+p(y_{i-1})\cdot t(y_{1+i \bmod l+1}, y_{\frac{l}{2}+i-1 \bmod l+1})$$

All shortest paths that have an endpoint in $y_i$ are part of the subtraction. This has to be done since $y_i$ has to lie between the endpoints of the shortest paths and may not be an endpoint. Next, we have to add all shortest paths where $y_{i-1}$ is an endpoint. These shortest paths were not included in $C_B^B(y_{i-1})$, because $y_{i-1}$ was an endpoint of them. But since $y_i$ is not an endpoint from the shortest paths going out from $y_{i-1}$ anymore, we now have to include these shortest paths. We need the biggest $i_1$, such that $y_i$ is still included in the shortest path from $y_{i-1}$ to $y_{i_1}$ (clockwise). Therefore, we solve the following inequation. The value $i_1-(i-1)$ is the distance from $i-1$ to $i_1$ (clockwise). Again, this value must not be greater than $\frac{\ell}{2}$ to include $y_i$:

$$i_1-(i-1) \le \frac{l}{2}$$
$$\iff \qquad i_1 \le \frac{l}{2}+i-1.$$

Note, that $i-1<i<i_1\le \frac{l}{2}+i-1$. Hence, $y_i$ lies between every shortest path from $y_{i-1}$ to $y_{i_1}$. Since the computations are correct for any $C_B^B(y_i)$, given that $C_B^B(y_{i-1})$ is correct, and the induction basis states that $C_B^B(y_0)$ is correct, our proof is now complete.

**Running time** The construction of $\mathrm{t_{right}}$ and $\mathrm{t_{left}}$ takes $O(\ell)$ time, if done as in (2). The evaluation of the formula $C_B^B(y_0)$ takes $O(\ell)$ time, too. The computation of any $C_B^B(y_i)$, $i\neq 0$, needs $O(1)$ time. Thus, the total running time per balloon is $O(\ell)$, $\sum_{B\in\mathcal{B}} O(|B|)$ for all balloons. Since there cannot be more balloons than vertices in the graph, it holds that $\sum_{B\in\mathcal{B}} O(|B|) = O(n)$. □

# 5 Chains: Dealing with paths

In this part, we will process all chains in $G$. In Section 3 we already mentioned that we will run Brandes' modified algorithm on the graph before processing the chains. In the next paragraphs, we will explain some details on Brandes' algorithm and show how we make use of the algorithm.

Basically, Brandes' algorithm performs a breadth first search (bfs) from every vertex of the graph. Doing one iteration, i. e. one bfs, on any vertex $s$ pairs $s$ with every other vertex $t$ and constructs every shortest path between $s$ and $t$. After the iteration the respective amount of betweenness centrality is added to all vertices $v \in V$ on these shortest paths. Since we use the modified version of Brandes' algorithm due to Baglioni et al. [Bag+12], the weight $p$ of each vertex is included in the computations. Thus, the sum computed by one such bfs is

$$\sum_{\substack{t \in V \\ s \neq v \neq t}} p(s) \cdot p(t) \frac{\sigma_{st}(v)}{\sigma_{st}}.$$

We restrict Brandes' algorithm to vertices of degree at least three, that is, only performing a bfs from each $s \in \mathrm{V}^{\geq 3}$ and only considering the shortest paths going out from these $s$. If we do this for all $s \in \mathrm{V}^{\geq 3}$, then we compute the sum

$$\sum_{\substack{s \in \mathrm{V}^{\geq 3}, t \in V \\ s \neq v \neq t}} p(s) \cdot p(t) \frac{\sigma_{st}(v)}{\sigma_{st}}.$$

This sum is computed in line 17 of Algorithm 1.

Afterwards, we still have to consider the shortest paths going out from the vertices of degree two. We have to compute the sum

$$\sum_{\substack{s \in \mathrm{V}^{=2}, t \in V \\ s \neq v \neq t}} p(s) \cdot p(t) \frac{\sigma_{st}(v)}{\sigma_{st}}.$$

As all vertices of degree at most one were already deleted, combining both sums yields the correct betweenness centrality

$$\sum_{\substack{s,t \in V \\ s \neq v \neq t}} p(s) \cdot p(t) \frac{\sigma_{st}(v)}{\sigma_{st}}.$$

This allows us to treat the vertices of degree two separately. Instead of performing one bfs on every vertex in $\mathrm{V}^{=2}$, we will only perform one bfs on each endpoint of each chain. These bfs' are performed when running Brandes' algorithm on $G$, since the endpoint of the chains are of degree at least three. Afterwards we do some extra computations on the chains and will process all vertices of a chain all together. This is faster than performing a bfs on each vertex of a chain separately, if the chains are large enough. The bigger the chains are the bigger the speed up is.

While we perform Brandes' algorithm on $G$, we will store some additional information. These information will help when processing the chains later. For each chain $C = (x, y_1, ..., y_\ell, z) \in \mathcal{C}$ and each $v \in \mathrm{V}^{\geq 3}$ we store

- the distances $d_G(v, x)$ and $d_G(v, z)$;

- the number of shortest paths from $v$ to $x$ ($\sigma_{vx}$) and from $v$ to $z$ ($\sigma_{vz}$);
- the shortest path(s) from $v$ to $x$ and $z$, respectively. (These can be stored by storing the DAG that is created by running Brandes' algorithm from $x$ or $z$, respectively.)

### 5.1 Processing the chains

We assume that we already ran Brandes' algorithm on $G$. Thus, for all $v \in V$ we computed the sum

$$\sum_{\substack{s \in \mathrm{V}^{\geq 3}, t \in \mathrm{V}^{\geq 3} \\ s \neq v \neq t}} p(s) \cdot p(t) \frac{\sigma_{st}(v)}{\sigma_{st}}.$$

Now, we still have to calculate these sums:

$$\sum_{\substack{s \in \mathrm{V}^{=2}, t \in V \\ s \neq v \neq t}} p(s) \cdot p(t) \frac{\sigma_{st}(v)}{\sigma_{st}} \tag{6}$$

and

$$\sum_{\substack{s \in \mathrm{V}^{\geq 3}, t \in \mathrm{V}^{=2} \\ s \neq v \neq t}} p(s) \cdot p(t) \frac{\sigma_{st}(v)}{\sigma_{st}}. \tag{7}$$

Let us first look at the first of the two sums (6). Since all vertices of degree two are part of exactly one chain, the following is true (Recall, that $\mathcal{C}$ is the set of all chains in $G$.):

$$\sum_{\substack{s \in \mathrm{V}^{=2}, t \in V \\ s \neq v \neq t}} p(s) \cdot p(t) \frac{\sigma_{st}(v)}{\sigma_{st}} = \sum_{\substack{s \in S^*(C), t \in V \\ C \in \mathcal{C} \\ s \neq v \neq t}} p(s) \cdot p(t) \frac{\sigma_{st}(v)}{\sigma_{st}}. \tag{8}$$

**Splitting the sum** We separate the computation of the above sum (8) into three steps. In every step we handle the shortest paths between different pairs of vertices. The three steps are:

**Step 1** Shortest paths between each chain and each vertex of degree at least three (9)

**Step 2** Shortest paths between each pair of two chains (10)

**Step 3** Shortest paths between each two vertices in a single chain (11)

Splitting the sum will result in the following sub sums, each of them being computed in the respective step above:

$$\sum_{\substack{s \in S^*(C), t \in V \\ C \in \mathcal{C} \\ s \neq v \neq t}} p(s) \cdot p(t) \frac{\sigma_{st}(v)}{\sigma_{st}} =$$

$$\sum_{\substack{s \in S^*(C), t \in \mathrm{V}^{\geq 3} \\ C \in \mathcal{C} \\ s \neq v \neq t}} p(s) \cdot p(t) \frac{\sigma_{st}(v)}{\sigma_{st}} \tag{9}$$

$$+ \sum_{\substack{s \in S^*(C_1), t \in S^*(C_2) \\ C_1, C_2 \in \mathcal{C} \\ C_1 \neq C_2 \\ s \neq v \neq t}} p(s) \cdot p(t) \frac{\sigma_{st}(v)}{\sigma_{st}} \tag{10}$$

$$+ \sum_{\substack{s,t \in S^*(C) \\ C \in \mathcal{C} \\ s \neq v \neq t}} p(s) \cdot p(t) \frac{\sigma_{st}(v)}{\sigma_{st}} \tag{11}$$

Each of the above steps and its respective sub sum is computed in a separate sub section, in the given order.

Before we show to compute (9) - (11), look at (7). This sum still needs to be considered. Therefore, we will make use of the symmetry of shortest paths:

$$\sum_{\substack{s \in \mathrm{V}^{\geq 3}, t \in \mathrm{V}^{=2} \\ s \neq v \neq t}} p(s) \cdot p(t) \frac{\sigma_{st}(v)}{\sigma_{st}} = \sum_{\substack{s \in \mathrm{V}^{=2}, t \in \mathrm{V}^{\geq 3} \\ s \neq v \neq t}} p(s) \cdot p(t) \frac{\sigma_{st}(v)}{\sigma_{st}} = \sum_{\substack{s \in S^*(C), t \in \mathrm{V}^{\geq 3} \\ C \in \mathcal{C} \\ s \neq v \neq t}} p(s) \cdot p(t) \frac{\sigma_{st}(v)}{\sigma_{st}} \tag{12}$$

This sum is already computed in Step 1 (9). Thus, we just need to multiply the results of Step 1 by two to also include this sum.

For each $C = (x, y_1, ..., y_\ell, z) \in \mathcal{C}$ we construct two tables $\mathrm{t_{left}}$ and $\mathrm{t_{right}}$ similar to the tables constructed for each balloon in Section 4. These tables store the amount of vertices that were connected to the vertices $y_1, ..., y_i$ or $y_i, ..., y_{\ell_1}$, respectively. We will make use of them in the following subsections:

$$\mathrm{t_{left}}(y_i) = \begin{cases} 0, \text{ if } i < 1 \text{ or } i > \ell \\ p(y_1), \text{ if } i = 1 \\ \mathrm{t_{left}}(y_{i-1}) + p(y_i), \text{ otherwise} \end{cases}$$

$$\mathrm{t_{right}}(y_i) = \begin{cases} 0, \text{ if } i < 1 \text{ or } i > \ell \\ p(y_\ell), \text{ if } i = \ell \\ \mathrm{t_{right}}(y_{i+1}) + p(y_i), \text{ otherwise} \end{cases} .$$

## 5.2 Step 1: Shortest paths between chains and vertices of degree greater than two

In this step, for every $v \in V$, we calculate sum (9) :

$$\sum_{\substack{s \in S^*(C), t \in \mathrm{V}^{\geq 3} \\ C \in \mathcal{C} \\ s \neq v \neq t}} \frac{\sigma_{st}(v)}{\sigma_{st}}.$$

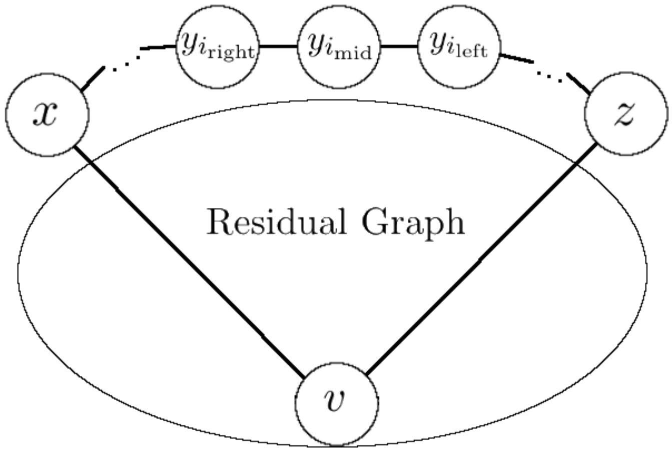



Figure 6: Chain $C$ and vertex $v$

The respective line in Algorithm 1 is line 22.

We will look at all shortest paths between each chain and each vertex of degree at least two. On these shortest paths, there are vertices which are part of the chain, i. e. which are inside the chain, and there are vertices which are outside the chain. We split our computations into two parts. In the first part we will show how to increase the betweenness centrality of the vertices outside the chain. In the second part we will look at the vertices inside the chain.

**Vertices outside the chain** For every chain $C = (x, y_1, ..., y_\ell, z) \in \mathcal{C}$ we look at every $v \in V \setminus S^*(C) = V \setminus \{y_1, ..., y_\ell\}$. From every $y_i$, $1 \leq i \leq \ell$, there are some shortest paths to $v$, each of them either passing through $x$ or $z$. To determine the exact number of shortest paths to $v$ through $x$ and $z$, respectively, we have to determine the "middle index" of chain $C$. That is $i_{\text{mid}} \in \mathbb{Q}$, $0 \leq i_{\text{mid}} \leq \ell$, such that the shortest paths from all vertices $y_i$ with $i < i_{\text{mid}}$ to $v$ pass through $x$ and the shortest paths from all vertices $y_i$ with $i > i_{\text{mid}}$ to $v$ pass through $z$. If $i_{\text{mid}}$ is an integer, then the vertex $y_{i_{\text{mid}}}$ exists. Then, from $y_{i_{\text{mid}}}$ there are some shortest paths to $v$, which pass $x$ and some, which pass $z$. Otherwise, there is no vertex $y_{i_{\text{mid}}}$. Additionally, let $i_{\text{left}}$ be the largest integer such that $i_{\text{left}} < i_{\text{mid}}$ and let $i_{\text{right}}$ be the smallest integer such that $i_{\text{right}} > i_{\text{mid}}$. See Figure 6 for an illustration.

To calculate $i_{\text{mid}}$, we solve the following equation:

$$d_G(v, x) + i_{\text{mid}} = d_G(v, z) + \ell + 1 - i_{\text{mid}}$$

$$\iff \qquad i_{\text{mid}} = \frac{d_G(v, z) + \ell + 1 - d_G(v, x)}{2}$$

The term $d_G(v, x) + i_{\text{mid}}$ is the length of the path from $v$ to $i_{\text{mid}}$ entering $C$ via $x$, while $d_G(v, z) + \ell + 1 - i_{\text{mid}}$ is the length of the path from $v$ to $i_{\text{mid}}$ entering $C$ via $z$. We basically determine the middle of the chain and then shift that middle to the left (or right) by half of the difference of the distances between $v$ and $x$ and $v$ and $z$. If $i_{\text{mid}}$ should be smaller than one or greater than $\ell$, we set it to one or $\ell$, respectively.

Recall that we stored the amount of shortest paths from $x$ and $z$ to $v$ and the shortest paths itself in Section 5.1. From $x$ to $v$, there are $\sigma_{xv}$ shortest paths and from $z$ to $v$, there are $\sigma_{zv}$ shortest paths. From each $y_i$ there are $\sigma_{xv}$ or $\sigma_{zv}$ shortest paths to $v$, too, depending on wheter $i \leq i_{\text{left}}$ or $i \geq i_{\text{right}}$, because between each $y_i$ and $x$ or $z$, there is exactly one shortest path. We also want to take into account the vertices that were originally connected to $v$ and each $y_i$. There were $p(v)$ vertices originally connected to $v$ (including $v$) and there were $p(y_i)$ vertices originally connected to each $y_i$ (including $y_i$). When we sum up all $p(y_i)$, we get $\text{t}_{\text{left}}(y_{i_{\text{left}}})$ if $i \leq i_{\text{left}}$, or $\text{t}_{\text{right}}(y_{i_{\text{right}}})$ if $i \geq i_{\text{right}}$. For each vertex $w$ lying on a shortest path between $x$ and $v$ or $z$ and $v$, increase its betweenness centrality by

$$\sum_{1 \leq i \leq i_{\text{left}}} p(i)p(v)\frac{\sigma_{xv}(w)}{\sigma_{xv}} = \text{t}_{\text{left}}(y_{i_{\text{left}}}) \cdot p(v)\frac{\sigma_{xv}(w)}{\sigma_{xv}}$$

or

$$\sum_{i_{\text{right}} \leq i \leq \ell} p(i)p(v)\frac{\sigma_{zv}(w)}{\sigma_{zv}} = \text{t}_{\text{right}}(y_{i_{\text{right}}}) \cdot p(v)\frac{\sigma_{zv}(w)}{\sigma_{zv}},$$

respectively.

If $y_{i_{\text{mid}}}$ exists, then we add a value of

$$p(y_{i_{\text{mid}}})p(v)\frac{\sigma_{xv}(w) + \sigma_{zv}(w)}{\sigma_{xv} + \sigma_{zv}} \tag{13}$$

to every $w \in \text{V}^{\geq 3}$ which lies on a path to from $x$ to $v$ or on a path from $z$ to $v$.

**Vertices inside the chain** The above computations only increase the betweenness centrality of vertices outside the chain. For vertices in $C$ the computations are different. Each $y_i \in S^*(C)$ is included in all shortest paths from vertices between $y_i$ and $y_{i_{\text{mid}}}$ to $v$. Thus, for $y_i$, we have to increase its betweenness centrality by

$$(\text{t}_{\text{left}}(y_{i_{\text{left}}}) - \text{t}_{\text{left}}(i)) \cdot p(v)$$

if $i < i_{\text{mid}}$, and by

$$(\text{t}_{\text{right}}(y_{i_{\text{right}}}) - \text{t}_{\text{right}}(i)) \cdot p(v)$$

if $i > i_{\text{mid}}$.

If $y_{i_{\text{mid}}}$ exists, then we additionally have to increase the betweenness centrality of each $y_i$ by

$$p(y_{i_{\text{mid}}}) \cdot p(v)\frac{\sigma_{xv}}{\sigma_{xv} + \sigma_{zv}}$$

if $i < i_{\text{mid}}$, and by

$$p(y_{i_{\text{mid}}}) \cdot p(v)\frac{\sigma_{zv}}{\sigma_{xv} + \sigma_{zv}}$$

if $i > i_{\text{mid}}$, since there are paths from $y_{i_{\text{mid}}}$ to $v$ either passing through $x$ or $z$ and each $y_i$ can be included in only one of these paths. If $i = i_{\text{mid}}$, then we do not increase its betweenness centrality at all, because there are no shortest paths passing through $y_{i_{\text{mid}}}$.

We now have considered each shortest path between $v$ and vertices $y_i$ of chain $C$ and increased the betweenness of every vertex $w$ lying on these paths by their respective amount.

As shown in (12), all results have to be added twice to the betweenness centrality of the respective vertex. Next, we proof the correctness of the computations done in Step 1 by proving the following lemma:

**Lemma 5.1.** *In Step 1 we compute for each $v \in V$*

$$\sum_{\substack{s \neq v \neq t \\ s \in C, t \in \mathrm{V}^{\geq 3} \\ C \in \mathcal{C}}} \frac{\sigma st(v)}{\sigma st}.$$

*Proof.* We will split the proof into two parts. In the first, we show the proof for the vertices outside the chain. In the second part, we show the proof for the vertices inside the chain.

**Vertices outside the chain** Before beginning with the proof, we define the following:

$$f_v(w, i_{\text{mid}}) := \begin{cases} p(y_{i_{\text{mid}}})p(v)\frac{\sigma_{xv}(w)+\sigma_{zv}(w)}{\sigma_{xv}+\sigma_{zv}}, & \text{if } y_{i_{\text{mid}}} \text{ exists} \\ 0, & \text{otherwise} \end{cases}$$

The function $f_v$ represents the amount of betweenness that needs to be added to vertex $w \in V \setminus S^*(V) \setminus \{v\}$ according to (13), depending of whether $y_{i_{\text{mid}}}$ exists or not. For $f_v$, the following holds (if $y_{i_{\text{mid}}}$ exists):

$$f_v(w, i_{\text{mid}}) = p(y_{i_{\text{mid}}})p(v)\frac{\sigma_{xv}(w) + \sigma_{zv}(w)}{\sigma_{xv} + \sigma_{zv}} = p(y_{i_{\text{mid}}})p(v)\frac{\sigma_{y_{i_{\text{mid}}}v}(w)}{\sigma_{y_{i_{\text{mid}}}v}}$$

This is true, since from $y_{i_{\text{mid}}}$ to $v$ there are shortest path through both $x$ and $z$. Thus, the number of shortest paths from $y_{i_{\text{mid}}}$ to $v$ is the number of shortest paths from $x$ to $v$ plus the number of shortest paths from $z$ to $v$.

For each $C = (x, y_1, ..., y_\ell, z) \in \mathcal{C}$, each $v \in \mathrm{V}^{\geq 3}$ and each $w \in V \setminus S^*(C) \setminus \{v\}$ we compute the following term in Step 1:

$$
\begin{aligned}
&\mathrm{t}_{\mathrm{left}}(y_{i_{\mathrm{left}}}) \cdot p(v) \frac{\sigma_{xv}(w)}{\sigma_{xv}} + \mathrm{t}_{\mathrm{right}}(y_{i_{\mathrm{right}}}) \cdot p(v) \frac{\sigma_{zv}(w)}{\sigma_{zv}} + f_v(w, i_{\mathrm{mid}}) \\
&\overset{\text{(i)}}{=} \sum_{1 \le k \le i_{\mathrm{left}}} p(y_k) p(v) \frac{\sigma_{xv}(w)}{\sigma_{xv}} + \sum_{i_{\mathrm{right}} \le k \le \ell} p(y_k) p(v) \frac{\sigma_{zv}(w)}{\sigma_{zv}} + f_v(w, i_{\mathrm{mid}}) \\
&\overset{\text{(ii)}}{=} \sum_{1 \le k \le i_{\mathrm{left}}} p(y_k) p(v) \frac{\sigma_{y_k v}(w)}{\sigma_{y_k v}} + \sum_{i_{\mathrm{right}} \le k \le \ell} p(y_k) p(v) \frac{\sigma_{y_k v}(w)}{\sigma_{y_k v}} + f_v(w, i_{\mathrm{mid}}) \\
&= \sum_{1 \le k \le \ell} p(y_k) p(v) \frac{\sigma_{y_k v}(w)}{\sigma_{y_k v}} \\
&= \sum_{y \in S^*(C)} p(y) p(v) \frac{\sigma_{yv}(w)}{\sigma_{yv}}
\end{aligned}
$$

(i): by construction of $\mathrm{t}_{\mathrm{left}}$ and $\mathrm{t}_{\mathrm{right}}$

(ii): From each $y_k$, $1 \le k \le i_{\mathrm{left}}$, there is one shortest path to $x$. From $x$ there are $\sigma_{xv}$ shortest paths to $v$. Thus, from each $y_k$, there are $\sigma_{xv}$ shortest paths to $v$, too. So, $\sigma_{xv} = \sigma_{y_k v}$ is true. Since these shortest paths also include the same vertices outside the chain and $w$ is outside the chain, too, $\sigma_{xv}(w) = \sigma_{y_k v}(w)$ is also true. For each $y_k$, $i_{\mathrm{right}} \le k \le \ell$, this works analogously.

We do this for each $C \in \mathcal{C}$ and for each $v \in \mathrm{V}^{\ge 3} \setminus \{w\}$:

$$
\sum_{C \in \mathcal{C}} \sum_{v \in \mathrm{V}^{\ge 3}} \sum_{y \in S^*(C)} p(y) p(v) \frac{\sigma_{yv}(w)}{\sigma_{yv}} = \sum_{\substack{C \in \mathcal{C} \\ y \in S^*(C), v \in \mathrm{V}^{\ge 3} \\ y \ne w \ne v}} p(y) p(v) \frac{\sigma_{yv}(w)}{\sigma_{yv}}
$$

Because $y \in \mathrm{V}^{=2}$ and $w \in \mathrm{V}^{\ge 3}$, $y \ne w$ is true. Since $v \in \mathrm{V}^{\ge 3} \setminus \{w\}$, $v \ne w$ is true, too.

**Vertices inside the chain** Now, we look at the vertices inside the chain, i. e. $y_i \in S^*(C)$. According to our procedure in Step 1, for each $C = (x, y_1, ..., y_\ell, z) \in \mathcal{C}$ and $v \in \mathrm{V}^{\ge 3}$, we have to increase the betweenness centrality of $y_i$, $1 \le i \le i_{\mathrm{left}}$, by the following value

(The proof for $i_{\text{right}} \le i \le \ell$ works analogously.):

$$\begin{aligned}
&(\mathrm{t}_{\text{left}}(y_{i_{\text{left}}}) - \mathrm{t}_{\text{left}}(y_i))p(v) + f_v(w, i_{\text{mid}}) \\
&= (\sum_{1 \le k \le i_{\text{left}}} p(y_k) - \sum_{1 \le k \le i} p(y_k))p(v) + f_v(w, i_{\text{mid}}) \\
&= (\sum_{i < k \le i_{\text{left}}} p(y_k))p(v) + f_v(w, i_{\text{mid}}) \\
&\overset{\text{(iii)}}{=} (\sum_{i < k \le i_{\text{left}}} p(y_k) \frac{\sigma_{y_k v}(y_i)}{\sigma_{y_k v}})p(v) + f_v(w, i_{\text{mid}}) \\
&\overset{\text{(iv)}}{=} (\sum_{i < k \le i_{\text{left}}} p(y_k) \frac{\sigma_{y_k v}(y_i)}{\sigma_{y_k v}} + \sum_{1 \le k < i} p(y_k) \frac{\sigma_{y_k v}(y_i)}{\sigma_{y_k v}} \\
&+ \sum_{i_{\text{right}} \le k \le \ell} p(y_k) \frac{\sigma_{y_k v}(y_i)}{\sigma_{y_k v}})p(v) + f_v(w, i_{\text{mid}}) \\
&= \sum_{\substack{1 \le k \le \ell \\ i \ne k}} p(y_k) \frac{\sigma_{y_k v}(y_i)}{\sigma_{y_k v}} p(v) \\
&= \sum_{\substack{y \in S^*(C) \\ y \ne y_i}} p(y)p(v) \frac{\sigma_{yv}(y)}{\sigma_{yv}}
\end{aligned}$$

(iii): From each $y_k$, $i < k \le i_{\text{left}}$, there are some shortest paths to $v$. These shortest paths all leave chain $C$ through $x$, which means that $y_i$ is part of these shortest paths. Hence, $\sigma_{y_k v}(y_i) = \sigma_{y_k v} = \frac{\sigma_{y_k v}(y_i)}{\sigma_{y_k v}} = 1$.

(iv): From each $y_k$, $1 < k < i$, there are some shortest paths to $v$. These shortest paths all leave chain $C$ through $x$. But since $k < i$, $y_i$ is not part of these shortest paths and $\sigma_{y_k v}(y_i) = 0$ and $\frac{\sigma_{y_k v}(y_i)}{\sigma_{y_k v}} = 0$.

From each $y_k$, $i_{\text{right}} \le k \le \ell$, the shortest paths to $v$ leave the chain through $z$. Thus, $y_i$ is not part of these shortest paths, which means that $\sigma_{y_k v}(y_i) = 0$ and $\frac{\sigma_{y_k v}(y_i)}{\sigma_{y_k v}} = 0$.

Again, we do this for each $C \in \mathcal{C}$ and for each $v \in \mathrm{V}^{\ge 3} \setminus \{w\}$:

$$\sum_{C \in \mathcal{C}} \sum_{v \in \mathrm{V}^{\ge 3}} \sum_{y \in S^*(C)} p(y)p(v) \frac{\sigma_{yv}(w)}{\sigma_{yv}} = \sum_{\substack{C \in \mathcal{C} \\ y \in S^*(C), v \in \mathrm{V}^{\ge 3} \\ y \ne w \ne v}} p(y)p(v) \frac{\sigma_{yv}(w)}{\sigma_{yv}}$$

**Running time** The running time of Step 1 depends on the number of chains and on the number of vertices that have a degree of at least three. For each chain and each vertex of degree at least three, we have to increase the betweenness centrality of each vertex lying on the shortest paths between the chain and the vertex. The number of the vertices lying on those shortest paths depends on $n$. Since we can bound the number of chains in the graph and the vertices of degree at least three in the graph by the feedback edge number $k$, the following holds: $O(|\mathcal{C}| \cdot |\mathrm{V}^{\geq 3}| \cdot n) = O(k^2 \cdot n)$ time. □

We will now continue with Step 2.

### 5.3 Step 2: Shortest paths between pairs of chains

In this section, we compute sum (10):

$$\sum_{\substack{C_1,C_2\in\mathcal{C}\\ s\in S^*(C_1),t\in S^*(C_2)\\ C_1\neq C_2\\ s\neq v\neq t}} p(s)\cdot p(t)\frac{\sigma_{st}(v)}{\sigma_{st}},$$

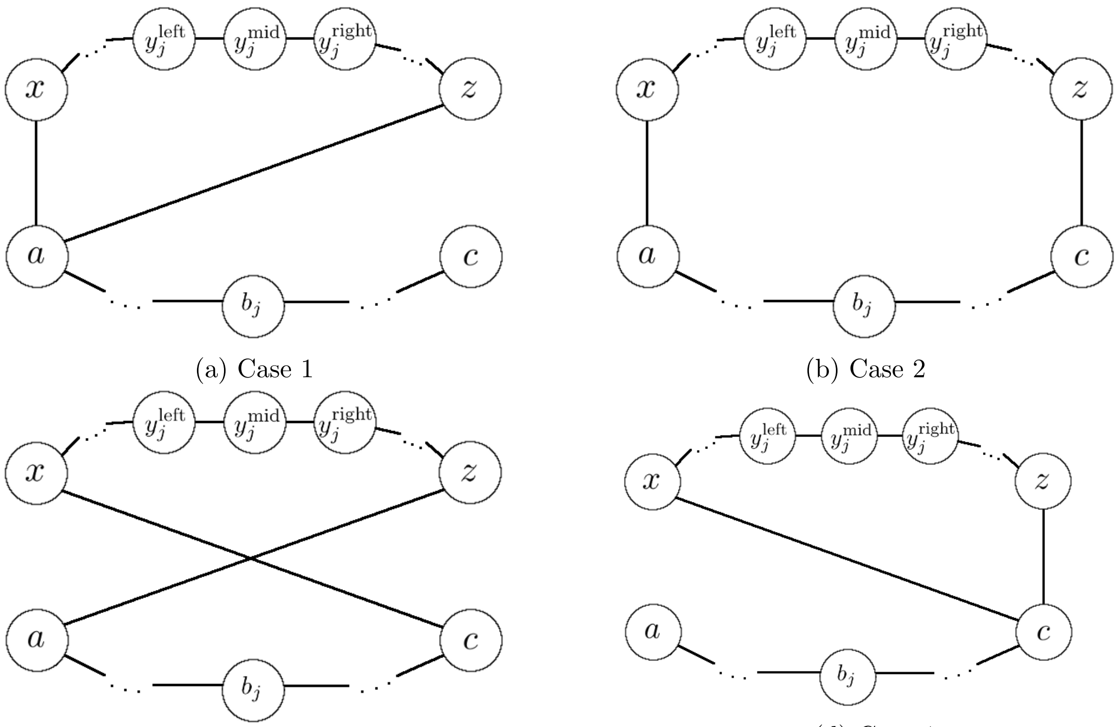
by considering all shortest paths between each pair of chains. This corresponds to line 26 in Algorithm 1.

As mentioned above every vertex of degree two is part of exactly one chain. Thus, if we compute all shortest paths between the vertices of each pair of chains and add the betweenness centrality to the vertices lying on these paths and do the same for all pairs of vertices within one chain, we have also considered all shortest paths between vertices of degree two. In this step, we will only look at the pairs of vertices between two chains.

Let $C_1 = (x, y_1, ..., y_{\ell_1}, z)$ and $C_2 = (a, b_1, ..., b_{\ell_2}, c)$ be any pair of two different chains and let $1 \leq i \leq \ell_1$ and $1 \leq j \leq \ell_2$. Also, let $I^j_{\text{mid}} = \frac{\ell_1}{2} - (d_G(x, b_j) - d_G(z, b_j))$ be the index such that all shortest paths from vertices $y_i$, $i < I^j_{\text{mid}}$, to $b_j$ leave $C_1$ through $x$ and all shortest paths from vertices $y_i$, $i > I^j_{\text{mid}}$, to $b_j$ leave $C_1$ through $z$. For easier reading, let $y^j_{\text{mid}} = y_{I^j_{\text{mid}}}$. The shortest paths from $y^j_{\text{mid}}$, if existing, to $b_j$ leave $C_1$ through $x$ or $z$. Additionally, let $I^j_{\text{left}}$ be the largest integer such that $I^j_{\text{left}} < I^j_{\text{mid}}$ and let $I^j_{\text{right}}$ be the smallest integer such that $I^j_{\text{right}} > I^j_{\text{mid}}$. Again, for easier reading, let $y^j_{\text{left}} = y_{I^j_{\text{left}}}$ and $y^j_{\text{right}} = y_{I^j_{\text{right}}}$.

Now look at a shortest path from any $y \in S^*(C_1)$ to any $b \in S^*(C_2)$. This shortest path can leave $C_1$ through either $x$ or $z$ and enter $C_2$ through either $a$ or $c$. This leads to four possible combinations of vertices that a shortest path from $y$ to $b$ has to pass. We distinguish these four combinations in four cases. In Figure 7 are the four cases. The lines between the vertices $x$ or $z$ and $a$ or $c$ represent the shortest paths between those vertices. In Figure 7a, for example, the shortest paths from vertices $y_i$ to $b_j$ leave $C_1$ through $x$ or $z$, respectively, depending of whether $i < I^j_{\text{mid}}$ or $i > I^j_{\text{mid}}$ and enter $C_2$ via $a$. We have to distinguish these cases, because in each of the four cases we have to do different computations. These computations follow in the next paragraphs. Now the cases follow:

- *Case 1* (see Figure 7a)
  The shortest paths from $x$ to $b_j$ and from $z$ to $b_j$ both enter $C_2$ through $a$.




(a) Case 1

(b) Case 2

(c) Case 3

(d) Case 4

Figure 7: The four possible cases

- *Case 2* (see Figure 7b)
  The shortest paths from $x$ to $b_j$ enter $C_2$ through $a$ while the shortest paths from $z$ to $b_j$ enter $C_2$ through $c$.

- *Case 3* (see Figure 7c)
  The shortest paths from $x$ to $b_j$ enter $C_2$ through $c$ while the shortest paths from $z$ to $b_j$ enter $C_2$ through $a$.

- *Case 4* (see Figure 7d)
  The shortest paths from $x$ to $b_j$ and from $z$ to $b_j$ both enter $C_2$ through $c$.

A case *applies* for a vertex $b_j$, $1 \leq j \leq \ell$, if the shortest paths from $x$ and $z$ to $b_j$ behave as described in the respective case. Depending on the case that applies for a given $b_j$, the case that applies for the following $b_{j'}$, $j' > j$, is restricted:

If for a vertex $b_j$ either *Case 2* or *3* applies, then for $b_{j'}$, $j' > j$, there can only apply *Case 4* (or still Case 2 or 3, respectively). If for a vertex $b_j$ *Case 4* applies, then for $b_{j'}$, $j' > j$, no other case applies. This is proven in Section 5.3.

We now group the vertices in $C_2$ by the case which applies for them. Since the order in which the cases apply is fixed we can calculate indices $j_1$, $j_2$, $1 \leq j_1 \leq j_2 \leq \ell_2 + 1$, such that for every $b_j$, $j < j_1$, Case 1 applies, for every $b_j$, $j_1 \leq j < j_2$, either Case 2 or 3 applies and for every $b_j$, $j_2 \leq j \leq \ell_2$, Case 4 applies.

We first compute $j_1$. For each $1 \leq j < j_1$ the length of the path from $b_j$ to $x$ by leaving $C_2$ via $a$ has to be smaller than the length of the path from $b_j$ to $x$ by leaving $C_2$ via $c$, and the length of the path from $b_j$ to $z$ by leaving $C_2$ via $a$ has to be smaller than the length of the path from $b_j$ to $z$ by leaving $C_2$ via $c$. This results in two inequalities to be true:

$$\begin{aligned} & d_G(a,x) + j < d_G(c,x) + \ell_2 + 1 - j \\ \iff \quad & j < \frac{d_G(c,x) + \ell_2 + 1 - d_G(a,x)}{2} \end{aligned}$$

and

$$\begin{aligned} & d_G(a,z) + j < d_G(c,z) + \ell_2 + 1 - j \\ \iff \quad & j < \frac{d_G(c,z) + \ell_2 + 1 - d_G(a,z)}{2} \cdot \end{aligned}$$

Both of the above inequalities result in a value for $j$. We need the smaller of the two results, thus:

$$j < j_1 = \min\{1, \frac{d_G(c,x) + \ell_2 + 1 - d_G(a,x)}{2}, \frac{d_G(c,z) + \ell_2 + 1 - d_G(a,z)}{2}\}.$$

The inequalities for $j_2$ are analogous, we just have to replace the smaller-than symbol by a greater-than symbol. So, for $j_2$, we get

$$j \geq j_2 = \min\{\ell_2, \max\{\frac{d_G(c,x) + \ell_2 + 1 - d_G(a,x)}{2}, \frac{d_G(c,z) + \ell_2 + 1 - d_G(a,z)}{2}\}\}.$$

For the different cases we have to do different computations. Again, we split these computations depending of whether the vertices are outside the chains or inside the chains:

### Vertices outside the chain (I)

**Case I.1** Recall, that $C_1 = (x, y_1, ..., y_{\ell_1}, z)$ and $C_2 = (a, b_1, ..., b_{\ell_2}, c)$. In this case, the shortest paths from the vertices $y_{k_1}$, $k_1 \leq I^j_{\text{left}}$, and $y_{k_2}$, $k_2 \geq I^j_{\text{right}}$, to each $b_j$, $1 \leq j < j_1$, leave $C_1$ through $x$ or $z$, respectively, and enter $C_2$ through $a$. Thus, between each $y_{k_1}$ or $y_{k_2}$ and each $b_j$ there are $\sigma_{xa}$ or $\sigma_{za}$ shortest paths, respectively. Since we also need to take into account the vertices that were originally connected to the respective vertices, we have to multiply these values by $p(b_j)$ and by $p(y_{k_1})$ or $p(y_{k_2})$. We have to increase the betweenness centrality of $w \in V$ lying on a shortest path from $x$ to $a$ by the following:

$$\sum_{1 \leq j < j_1} \sum_{1 \leq k \leq I^j_{\text{left}}} p(b_j) p(y_k) \frac{\sigma_{xa}(w)}{\sigma_{xa}} = \sum_{1 \leq j < j_1} \mathrm{t}_{\text{left}}(y^j_{\text{left}}) p(b_j) \frac{\sigma_{xa}(w)}{\sigma_{xa}}.$$

But for Case 1, $I^j_{\text{left}} = I^{j'}_{\text{left}}$ and $I^j_{\text{right}} = I^{j'}_{\text{right}}$ holds for any $j, j'$, $1 \leq j, j' < j_1$, since the difference of the distances $d_G(x, b_j)$ and $d_G(z, b_j)$ stays constant for all $1 \leq j < j_1$ (this also means that $y^j_{\text{left}} = y^{j'}_{\text{left}}$ and $y^j_{\text{right}} = y^{j'}_{\text{right}}$). Thus:

$$\sum_{1 \leq j < j_1} \mathrm{t}_{\text{left}}(y^j_{\text{left}}) p(b_j) \frac{\sigma_{xa}(w)}{\sigma_{xa}} = \mathrm{t}_{\text{left}}(y^{j_1-1}_{\text{left}}) \cdot \mathrm{t}_{\text{left}}(b_{j_1-1}) \frac{\sigma_{xa}(w)}{\sigma_{xa}}. \tag{14}$$

If $w \in V$ lies on a shortest path from $z$ to $a$, then the circumstances are analogous.

Summarizing our results, we have to increase the betweenness centrality of every $w \in V \setminus (S^*(C_1) \cup S^*(C_2))$ lying on a shortest path from $x$ to $a$ by

$$\mathrm{t}_{\mathrm{left}}(y_{\mathrm{left}}^{j_1-1}) \cdot \mathrm{t}_{\mathrm{left}}(b_{j_1-1}) \cdot \frac{\sigma_{xa}(w)}{\sigma_{xa}}$$

and the betweenness centrality of every $w$ lying on a shortest path from $z$ to $a$ by

$$\mathrm{t}_{\mathrm{right}}(y_{\mathrm{right}}^{j_1-1}) \cdot \mathrm{t}_{\mathrm{left}}(b_{j_1-1}) \cdot \frac{\sigma_{za}(w)}{\sigma_{za}}.$$

The distance from $y_{\mathrm{mid}}^j$, if existing, to $x$ is the same as to $z$. Thus, from this vertex there are $\sigma_{xa}$ shortest paths to $b_j$ leaving $C_1$ through $x$ and $\sigma_{za}$ shortest paths to $b_j$ leaving $C_1$ through $z$. For each vertex $w$ lying on at least one of these paths we have to increase its betweenness centrality by

$$p(y_{\mathrm{mid}}^{j_1-1}) \cdot \mathrm{t}_{\mathrm{left}}(b_{j_1-1}) \frac{\sigma_{xa}(w) + \sigma_{za}(w)}{\sigma_{xa} + \sigma_{za}}. \tag{15}$$

**Case I.2 (and I.3)** If Case 2 applies, between the vertices $y_{k_1}$, $k_1 \leq I_{\mathrm{left}}^j$ and $y_{k_2}$, $k_2 \geq I_{\mathrm{right}}^j$ and each $b_j$, $j_1 \leq j < j_2$, there are $\sigma_{xa}$ or $\sigma_{zc}$ shortest paths, respectively, because the shortest paths from $y_{k_1}$ to $b_j$ leave $C_1$ through $x$ and enter $C_2$ via $a$ while the shortest paths from $y_{k_2}$ to $b_j$ leave $C_1$ through $z$ and enter $C_2$ via $c$

This time, we have $I_{\mathrm{left}}^j \neq I_{\mathrm{left}}^{j'}$ and $I_{\mathrm{right}}^j \neq I_{\mathrm{right}}^{j'}$ for $j \neq j'$. This means, that we have to create a sum over all $b_j$ and cannot replace this sum by a single term as done in equation (14) in Case 1 above. We have to increase the betweenness centrality of every $w \in V \setminus (S^*(C_1) \cup S^*(C_2))$ lying on a shortest path from $x$ to $a$ by

$$\sum_{j_1 \leq k < j_2} \mathrm{t}_{\mathrm{left}}(y_{\mathrm{left}}^k) \cdot p(b_k) \cdot \frac{\sigma_{xa}(w)}{\sigma_{xa}}$$

and the betweenness centrality of every $w$ lying on a shortest path from $z$ to $c$ by

$$\sum_{j_1 \leq k < j_2} \mathrm{t}_{\mathrm{right}}(y_{\mathrm{right}}^k) \cdot p(b_k) \cdot \frac{\sigma_{zc}(w)}{\sigma_{zc}}$$

Also, if $y_{\mathrm{mid}}^j$ exists, we have to increase the betweenness centrality of each $w$ lying on a shortest paths from $x$ to $a$ or from $z$ to $c$ by

$$\sum_{j_1 \leq k < j_2} p(y_{\mathrm{mid}}^k) \cdot p(b_k) \cdot \frac{\sigma_{xa}(w) + \sigma_{zc}(w)}{\sigma_{xa} + \sigma_{zc}}. \tag{16}$$

Case I.3 is analogous to Case I.2: We only have to replace all appearances of $\sigma_{xa}$ in the above formulas by $\sigma_{xc}$ and all appearances of $\sigma_{zc}$ by $\sigma_{za}$. This needs to be done, since in Case 3 all shortest paths from vertices on the left of $y_{\mathrm{mid}}^j$ to $b_j$ enter $C_2$ via $c$ instead via $a$, and all shortest paths from vertices on the right of $y_{\mathrm{mid}}^j$ to $b_j$ enter $C_2$ via $a$ instead via $c$.

**Case I.4** Case I.4 is analogous to Case I.1: In the formulas of Case I.1, we only need to replace $j_1 - 1$ by $j_2$ and $a$ by $c$. This has to be done because in Case 4 all shortest paths leaving $C_1$ enter $C_2$ through $c$ (instead of $a$ as in Case 1).

**Vertices inside the chain (II)** In the above case distinction we only increased the betweenness centrality of vertices outside both chains. For the vertices inside the chains we have to do slightly different computations in the particular cases. We only increase the betweenness centrality of the vertices in $C_2$ and then double this value. Because we look at each pair of chains twice (with switched $C_1$ and $C_2$), this still produces correct results.

**Case II.1** As already mentioned in Case I.1 above between each $y \in C_1$ and each $b_j$, $1 \leq j < j_1$, there are shortest paths which all enter $C_2$ via $a$. Hence, any $b_{j'}$, $j' < j$, lies on all these shortest paths. The total amount of shortest paths that any $b_{j'}$ lies on is the number of vertices in $C_1$ (plus those, who were originally connected to vertices in $C_1$) multiplied by the number of vertices between $b_{j'}$ and $b_{j_1}$ (plus those, who were originally connected to them). This is

$$\sum_{j'<k<j_1} p(b_k) \cdot \sum_{1 \leq k \leq \ell_1} p(y_k) = (\mathrm{t}_{\mathrm{left}}(b_{j_1-1}) - \mathrm{t}_{\mathrm{left}}(b_{j'})) \cdot \mathrm{t}_{\mathrm{left}}(y_{\ell_1}) \tag{17}$$

shortest paths in total. We have to increase the betweenness centrality of each such $b_{j'}$ by this amount.

**Case II.2 (and Case II.3)** All shortest paths from vertices $y_{k_1}$, $k_1 \leq I^j_{\mathrm{left}}$ to $b_j$, $j_1 \leq j < j_2$, enter $C_2$ via $a$ and all shortest paths from vertices $y_{k_2}$, $k_2 \geq I^j_{\mathrm{right}}$, to $b_j$ enter $C_2$ via $z$. The shortest paths from the vertex $y^j_{\mathrm{mid}}$ to $b_j$, if existing, enter $C_2$ through either $a$ or $c$ according to whether leaving $C_1$ via $x$ or $z$. We need to consider the vertices in $C_2$ differently, depending on whether their index is smaller than $j_1$ (Case II.2.1), greater than $j_2$ (Case II.2.2) or between $j_1$ and $j_2$ (Case II.2.3).

**Case II.2.1** If $1 \leq j' < j_1$, then $b_{j'}$ lies on all shortest paths that enter $C_2$ via $a$. Summing over all $b_j$, $j_1 \leq j < j_2$, the total amount of these shortest paths is:

$$\sum_{k=j_1}^{j_2-1} \mathrm{t}_{\mathrm{left}}(y^k_{\mathrm{left}}) p(b_k). \tag{18}$$

We have to increase the betweenness centrality of each $b_{j'}$ by this value.

**Case II.2.2** If $j_2 \leq j' \leq \ell_2$, then $b_{j'}$ lies on all shortest paths that enter $C_2$ via $c$. Summing over all $b_j$, the total amount of those shortest paths is:

$$\sum_{k=j_1}^{j_2-1} \mathrm{t}_{\mathrm{right}}(y^k_{\mathrm{right}}) p(b_k). \tag{19}$$

Again, we have to increase the betweenness centrality of each $b_{j'}$ by this value.

**Case II.2.3** Now we look at each $b_{j'}$, $j_1 \leq j' < j_2$. Each $b_{j'}$ lies on all shortest paths that enter $C_2$ via $a$ and end in any $b_j$ with $j' < j < j_2$, and on all shortest paths that enter $C_2$ through $c$ and end in any vertex $b_j$ with $j_1 \leq j < j'$. The total amount of all these shortest paths is:

$$\sum_{k=j+1}^{j_2-1} \mathrm{t}_{\mathrm{left}}(y^k_{\mathrm{left}})p(b_k) + \sum_{k=j_1}^{j-1} \mathrm{t}_{\mathrm{right}}(y^k_{\mathrm{right}})p(b_k), \tag{20}$$

which is the value by which the betweenness centrality of each $b_{j'}$ has to be increased.

If $y^j_{\mathrm{mid}}$ exists, then we additionally have to increase the betweenness centrality of each $b_{j'}$, $1 \leq j' \leq \ell_2$, by

$$\sum_{j_1 \leq k < j_2} p(y^k_{\mathrm{mid}})p(b_k)\frac{\sigma_{y^k_{\mathrm{mid}} b_k}(b_{j'})}{\sigma_{y^k_{\mathrm{mid}} b_k}}$$

Case 3 is analogous.

**Case 4** This case is analogous to Case 1. We increase the betweenness centrality of each $b_j$, $j_2 \leq j \leq l_2$, by

$$2 \cdot (\mathrm{t}_{\mathrm{right}}(b_{j_2}) - \mathrm{t}_{\mathrm{right}}(b_j)) \cdot \mathrm{t}_{\mathrm{left}}(y_{\ell_1}). \tag{21}$$

We now increased the betweenness centrality of each vertex lying between any two chains. Next, we proof the correctness of Step 2 by proving the following lemma:

**Lemma 5.2.** *In Step 2 we compute for each $v \in V$*

$$\sum_{\substack{C_1, C_2 \in \mathcal{C} \\ s \in S^*(C_1), t \in S^*(C_2) \\ C_1 \neq C_2 \\ s \neq v \neq t}} \frac{\sigma_{st}(v)}{\sigma_{st}}.$$

*Proof.* Between every vertex in $C_1$ and each $b \in S^*(C_2)$ there are some shortest paths. We first show that we can distinguish these shortest paths through the four cases mentioned in Step 2. Next, we show that there are only two possible orders in which the cases can appear. Last, we show that the computations in the respective cases are correct.

The shortest paths from $C_1$ to any $b$ can start in one vertex $y_{k_1}$, $k_1 \leq I_{\mathrm{left}}$, $y_{k_2}$, $k_2 \geq I_{\mathrm{right}}$ or in $y_{\mathrm{mid}}$. Every shortest path from $y_{k_1}$ to $b$ leaves $C_1$ through $x$ and enters $C_2$ through either $a$ or $c$ and every shortest path from $y_{k_2}$ to $b$ leaves $C_1$ through $z$ and enters $C_2$ through either $a$ or $c$. The shortest paths from $y_{\mathrm{mid}}$, if existing, can both leave $C_1$ via $x$ and $z$ and enter $C_2$ via $a$ or $c$. This results in the four possible combinations already mentioned:

- *Case 1*: The shortest paths from $y$ to $b$ which leave $C_1$ through $x$ enter $C_2$ through $a$ and the shortest paths which leave $C_1$ through $z$ enter $C_2$ through $a$, too.

- *Case 2*: The shortest paths from $y$ to $b$ which leave $C_1$ through $x$ enter $C_2$ through $a$ and the shortest paths which leave $C_1$ through $z$ enter $C_2$ through $c$.
- *Case 3*: The shortest paths from $y$ to $b$ which leave $C_1$ through $x$ enter $C_2$ through $c$ and the shortest paths which leave $C_1$ through $z$ enter $C_2$ through $a$.
- *Case 4*: The shortest paths from $y$ to $b$ which leave $C_1$ through $x$ enter $C_2$ through $c$ and the shortest paths which leave $C_1$ through $z$ enter $C_2$ through $c$, too.

We will now show that the cases that may apply for a vertex $b_{j'}$, $1 \leq j < j' \leq \ell_2$, are restricted, depending on the case that applied for $b_j$.

If *Case 2* applies for $b_j$, then for $b_{j'}$ there can only apply either, still, *Case 2* or *Case 4*. With increasing $j$ the distance $d_G(b_j, a)$ becomes greater and the distance $d_G(b_j, c)$ becomes smaller. Thus, only the shortest paths from $C_1$ to $C_2$ which enter $C_2$ through $a$ can be replaced by an even shorter path, because the shortest paths which enter $C_2$ through $c$ become smaller as $d_G(b_j, c)$ becomes smaller. The only case which then is possible is *Case 4*. For *Case 3*, the circumstances are analogous.

If *Case 4* applies for $b_j$, then for $b_{j'}$ there cannot apply any other case than *Case 4*: The distance $d_G(b_j, c)$ becomes smaller with increasing $j$. Thus, there cannot be any other shortest paths from $C_1$ to $b_j$ which are even shorter, because in Case 4 all shortest paths from $C_1$ to $b_j$ enter $C_2$ via $c$.

Summarized, for increasing indices, from *Case 1* we can transition into either *Case 2*, *3* or *4*. From *Case 2* or *3* we can only transition into *Case 4* and from *Case 4* we cannot transition into any other case. Thus, we can separate $C_2$ by indexes $j_1, j_2$, $j_1 \leq j_2$, such that for every $b_j$, $1 \leq j < j_1$ *Case 1* applies, for every $b_j$, $j_1 \leq j < j_2$, either *Case 2* or *3* applies and for every $b_j$, $j_2 \leq j \leq l_2$, *Case 4* applies.

We now show that we do the correct computations for calculating the betweenness centrality in the four cases for $v \in V$. We will begin with the proof for the vertices outside $C_1$ and $C_2$. First, we will look at each case separately. Then, we will sum up the results of the different cases and show the correctness of the computations we did in this section. Afterwards, we do the same for the vertices inside $C_1$ and $C_2$.

**Vertices outside the chain (I)** As done in the proof for Step 1, we first define a function $f$. This function represents the amount of betweenness that has to be added to $w$ for the shortest paths between $b_j \in C_2$ and $y^j_{\text{mid}} \in C_1$ in each of the three cases if $y^j_{\text{mid}}$ exists, according to the formulas (15) and (16).

$$f(w,j) := \begin{cases} p(y^1_{\text{mid}}) \cdot \mathrm{t}_{\text{left}}(b_{j_1-1}) \frac{\sigma_{xa}(w)+\sigma_{za}(w)}{\sigma_{xa}+\sigma_{za}}, & \text{if } y^j_{\text{mid}} \text{ exists and } 1 \leq j < j_1 \\ \sum_{j_1 \leq k < j_2} p(y^k_{\text{mid}}) \cdot p(b_k) \frac{\sigma_{xa}(w)+\sigma_{zc}(w)}{\sigma_{xa}+\sigma_{zc}}, & \text{if } y^j_{\text{mid}} \text{ exists and } j_1 \leq j < j_2 \\ p(y^{\ell_2}_{\text{mid}}) \cdot \mathrm{t}_{\text{left}}(b_{j_1-1}) \frac{\sigma_{xc}(w)+\sigma_{zc}(w)}{\sigma_{xc}+\sigma_{zc}}, & \text{if } y^j_{\text{mid}} \text{ exists and } j_2 \leq j \leq \ell_2 \\ 0, \text{ otherwise} \end{cases},$$

We already show some equalities for $f$, which are needed in the proofs of the four cases, following in the next paragraphs just after these equalities. If Case 1 applies for $j$, then:

$$f(w,j) = p(y^1_{\text{mid}}) \cdot \text{t}_{\text{left}}(b_{j_1-1}) \frac{\sigma_{xa}(w) + \sigma_{za}(w)}{\sigma_{xa} + \sigma_{za}} = \sum_{1 \le k < j_1} p(y^1_{\text{mid}}) p(b_k) \frac{\sigma_{y^1_{\text{mid}} b_k}(w)}{\sigma_{y^1_{\text{mid}} b_k}}$$

If Case 2 or 3 applies for $j$, then:

$$f(w,j) = \sum_{j_1 \le k < j_2} p(y^k_{\text{mid}}) \cdot p(b_k) \frac{\sigma_{xa}(w) + \sigma_{zc}(w)}{\sigma_{xa} + \sigma_{zc}} = \sum_{j_1 \le k < j_2} p(y^k_{\text{mid}}) \cdot p(b_k) \frac{\sigma_{y^k_{\text{mid}} b_k}(w)}{\sigma_{y^k_{\text{mid}} b_k}},$$

If Case 4 applies for $j$, then:

$$f(w,j) = p(y^{\ell_2}_{\text{mid}}) \cdot \text{t}_{\text{left}}(b_{j_1-1}) \frac{\sigma_{xc}(w) + \sigma_{zc}(w)}{\sigma_{xc} + \sigma_{zc}} = \sum_{1 \le k < j_1} p(y^{\ell_2}_{\text{mid}}) p(b_k) \frac{\sigma_{y^{\ell_2}_{\text{mid}} b_k}(w)}{\sigma_{y^{\ell_2}_{\text{mid}} b_k}},$$

Recall, that for all vertices $b_j$, $1 \le j < j_1$, Case 1 applies, for all vertices $b_j$, $j_1 \le j < j_2$, either Case 2 or 3 applies and for all vertices $b_j$, $j_2 \le j < \ell_2$, Case 4 applies.

**Case I.1** According to our computations in Step 2, we have to increase the betweenness centrality of each vertex $w$ outside $C_1$ and $C_2$ by:

$$
\begin{aligned}
& \mathrm{t}_{\mathrm{left}}(y^1_{\mathrm{left}}) \cdot \mathrm{t}_{\mathrm{left}}(b_{j_1-1}) \cdot \frac{\sigma_{xa}(w)}{\sigma_{xa}} + \mathrm{t}_{\mathrm{right}}(y^1_{\mathrm{right}}) \cdot \mathrm{t}_{\mathrm{left}}(b_{j_1-1}) \cdot \frac{\sigma_{za}(w)}{\sigma_{za}} + f(w,j) \\
\stackrel{\text{(i)}}{=} & \sum_{1 \le k_1 < j_1} \mathrm{t}_{\mathrm{left}}(y^1_{\mathrm{left}}) p(b_{k_1}) \frac{\sigma_{xa}(w)}{\sigma_{xa}} + \sum_{1 \le k_1 < j_1} \mathrm{t}_{\mathrm{right}}(y^1_{\mathrm{right}}) p(b_{k_1}) \frac{\sigma_{za}(w)}{\sigma_{za}} + f(w,j) \\
\stackrel{\text{(i)}}{=} & \sum_{1 \le k_1 < j_1} \sum_{1 \le k_2 \le I^1_{\mathrm{left}}} p(y_{k_2}) p(b_{k_1}) \frac{\sigma_{xa}(w)}{\sigma_{xa}} \\
& + \sum_{1 \le k_1 < j_1} \sum_{I^1_{\mathrm{right}} \le k_2 \le \ell_1} p(y_{k_2}) p(b_{k_1}) \frac{\sigma_{za}(w)}{\sigma_{za}} + f(w,j) \\
\stackrel{\text{(ii)}}{=} & \sum_{1 \le k_1 < j_1} \sum_{1 \le k_2 \le I^1_{\mathrm{left}}} p(y_{k_2}) p(b_{k_1}) \frac{\sigma_{y_{k_2} b_{k_1}}(w)}{\sigma_{y_{k_2} b_{k_1}}} \\
& + \sum_{1 \le k_1 < j_1} \sum_{I^1_{\mathrm{right}} \le k_2 \le \ell_1} p(y_{k_2}) p(b_{k_1}) \frac{\sigma_{y_{k_2} b_{k_1}}(w)}{\sigma_{y_{k_2} b_{k_1}}} + f(w,j) \\
= & \sum_{1 \le k_1 < j_1} \Big( \sum_{1 \le k_2 \le I^1_{\mathrm{left}}} p(y_{k_2}) p(b_{k_1}) \frac{\sigma_{y_{k_2} b_{k_1}}(w)}{\sigma_{y_{k_2} b_{k_1}}} \\
& + \sum_{I^1_{\mathrm{right}} \le k_2 \le \ell_1} p(y_{k_2}) p(b_{k_1}) \frac{\sigma_{y_{k_2} b_{k_1}}(w)}{\sigma_{y_{k_2} b_{k_1}}} \Big) + f(w,j) \\
= & \sum_{1 \le k_1 < j_1} \sum_{1 \le k_2 \le \ell_1} p(y_{k_2}) p(b_{k_1}) \frac{\sigma_{y_{k_2} b_{k_1}}(w)}{\sigma_{y_{k_2} b_{k_1}}} \\
= & \sum_{1 \le k_1 < j_1} \sum_{y \in S^*(C_1)} p(y) p(b_{k_1}) \frac{\sigma_{y b_{k_1}}(w)}{\sigma_{y b_{k_1}}}
\end{aligned}
$$

(i): by definition of $\mathrm{t}_{\mathrm{left}}$ and $\mathrm{t}_{\mathrm{right}}$

(ii): From each $y_k$, $1 \le k \le I^1_{\mathrm{left}}$, there is one shortest path to $x$ and from each $y_k$, $I^1_{\mathrm{right}} \le k \le \ell_1$, there is one shortest path to $z$. From $x$ there are $\sigma_{xa}$ shortest paths to $a$ and from $z$ there are $\sigma_{za}$ shortest paths to $a$. From $a$ there is one shortest path to $b_j$, $1 \le j < j_1$. Thus, from each $y_k$, there are $\sigma_{xa}$ or $\sigma_{za}$ shortest paths to $b_j$, too, respectively. So, $\sigma_{xa} = \sigma_{y_k b_j}$ ($1 \le k \le I^1_{\mathrm{left}}$) and $\sigma_{za} = \sigma_{y_k b_j}$ ($I^1_{\mathrm{right}} \le k \le \ell_1$) is true.

**Case I.2 (I.3 is analogous)** In Case I.2 (or I.3) we have to increase the betweenness centrality of each vertex $w$ outside $C_1$ and $C_2$ by:

$$
\begin{aligned}
&\sum_{k_1=j_1}^{j_2-1} \mathrm{t}_{\text{left}}(y_{\text{left}}^{k_1}) \cdot p(b_{k_1}) \cdot \frac{\sigma_{xa}(w)}{\sigma_{xa}} + \sum_{k_1=j_1}^{j_2-1} \mathrm{t}_{\text{right}}(y_{\text{right}}^{k_1}) \cdot p(b_{k_1}) \cdot \frac{\sigma_{zc}(w)}{\sigma_{zc}} + f(w,j) \\
&\overset{\text{(i)}}{=} \sum_{j_1 \le k_1 < j_2} \sum_{1 \le k_2 \le I_{\text{left}}^{k_1}} p(y_{k_2}) p(b_{k_1}) \frac{\sigma_{xa}(w)}{\sigma_{xa}} + \sum_{j_1 \le k_1 < j_2} \sum_{I_{\text{right}}^{k_1} \le k_2 \le \ell_1} p(y_{k_2}) p(b_{k_1}) \frac{\sigma_{zc}(w)}{\sigma_{zc}} \\
&\overset{\substack{\text{analog}\\ \text{to (ii)}}}{=} \sum_{j_1 \le k_1 < j_2} \sum_{1 \le k_2 \le I_{\text{left}}^{k_1}} p(y_{k_2}) p(b_{k_1}) \frac{\sigma_{y_{k_2} b_{k_1}}(w)}{\sigma_{y_{k_2} b_{k_1}}} \\
&+ \sum_{j_1 \le k_1 < j_2} \sum_{I_{\text{right}}^{k_1} \le k_2 \le \ell_1} p(y_{k_2}) p(b_{k_1}) \frac{\sigma_{y_{k_2} b_{k_1}}(w)}{\sigma_{y_{k_2} b_{k_1}}} + f(w,j) \\
&= \sum_{j_1 \le k_1 < j_2} \Big( \sum_{1 \le k_2 \le I_{\text{left}}^{k_1}} p(y_{k_2}) p(b_{k_1}) \frac{\sigma_{y_{k_2} b_{k_1}}(w)}{\sigma_{y_{k_2} b_{k_1}}} \\
&+ \sum_{I_{\text{right}}^{k_1} \le k_2 \le \ell_1} p(y_{k_2}) p(b_{k_1}) \frac{\sigma_{y_{k_2} b_{k_1}}(w)}{\sigma_{y_{k_2} b_{k_1}}} \Big) + f(w,j) \\
&= \sum_{j_1 \le k_1 < j_2} \sum_{1 \le k_2 \le \ell_1} p(y_{k_2}) p(b_{k_1}) \frac{\sigma_{y_{k_2} b_{k_1}}(w)}{\sigma_{y_{k_2} b_{k_1}}} \\
&= \sum_{j_1 \le k_1 < j_2} \sum_{y \in S^*(C_1)} p(y) p(b_{k_1}) \frac{\sigma_{y b_{k_1}}(w)}{\sigma_{y b_{k_1}}}
\end{aligned}
$$

**Case I.4** We have to increase the betweenness centrality of each vertex $w$ lying outside the chain by the following amount:

$$
\begin{aligned}
&\mathrm{t}_{\text{left}}(y^{j_2}_{\text{left}}) \cdot \mathrm{t}_{\text{right}}(b_{j_2}) \cdot \frac{\sigma_{xc}(w)}{\sigma_{xc}} + \mathrm{t}_{\text{right}}(y^{j_2}_{\text{right}}) \cdot \mathrm{t}_{\text{right}}(b_{j_2}) \cdot \frac{\sigma_{zc}(w)}{\sigma_{zc}} + f(w,j) \\
&\overset{\text{(i)}}{=} \sum_{j_2 \le k_1 \le \ell_2} \mathrm{t}_{\text{left}}(y^{j_2}_{\text{left}}) p(b_{k_1}) \frac{\sigma_{xc}(w)}{\sigma_{xc}} + \sum_{j_2 \le k_1 \le \ell_2} \mathrm{t}_{\text{right}}(y^{j_2}_{\text{right}}) p(b_{k_1}) \frac{\sigma_{zc}(w)}{\sigma_{zc}} + f(w,j) \\
&\overset{\text{(i)}}{=} \sum_{j_2 \le k_1 \le \ell_2} \sum_{1 \le k_2 \le I^{j_2}_{\text{left}}} p(y_{k_2}) p(b_{k_1}) \frac{\sigma_{xc}(w)}{\sigma_{xc}} \\
&+ \sum_{j_2 \le k_1 \le \ell_2} \sum_{I^{j_2}_{\text{right}} \le k_2 \le \ell_1} p(y_{k_2}) p(b_{k_1}) \frac{\sigma_{zc}(w)}{\sigma_{zc}} + f(w,j) \\
&\overset{\substack{\text{analog}\\ \text{to (ii)}}}{=} \sum_{j_2 \le k_1 \le \ell_2} \sum_{1 \le k_2 \le I^{j_2}_{\text{left}}} p(y_{k_2}) p(b_{k_1}) \frac{\sigma_{y_{k_2} b_{k_1}}(w)}{\sigma_{y_{k_2} b_{k_1}}} \\
&+ \sum_{j_2 \le k_1 \le \ell_2} \sum_{I^{j_2}_{\text{right}} \le k_2 \le \ell_1} p(y_{k_2}) p(b_{k_1}) \frac{\sigma_{y_{k_2} b_{k_1}}(w)}{\sigma_{y_{k_2} b_{k_1}}} + f(w,j) \\
&= \sum_{j_2 \le k_1 \le \ell_2} \Big( \sum_{1 \le k_2 \le I^{j_2}_{\text{left}}} p(y_{k_2}) p(b_{k_1}) \frac{\sigma_{y_{k_2} b_{k_1}}(w)}{\sigma_{y_{k_2} b_{k_1}}} \\
&+ \sum_{I^{j_2}_{\text{right}} \le k_2 \le \ell_1} p(y_{k_2}) p(b_{k_1}) \frac{\sigma_{y_{k_2} b_{k_1}}(w)}{\sigma_{y_{k_2} b_{k_1}}} \Big) + f(w,j) \\
&= \sum_{j_2 \le k_1 \le \ell_2} \sum_{1 \le k_2 \le \ell_1} p(y_{k_2}) p(b_{k_1}) \frac{\sigma_{y_{k_2} b_{k_1}}(w)}{\sigma_{y_{k_2} b_{k_1}}} \\
&= \sum_{j_2 \le k_1 \le \ell_2} \sum_{y \in S^*(C_1)} p(y) p(b_{k_1}) \frac{\sigma_{y b_{k_1}}(w)}{\sigma_{y b_{k_1}}}
\end{aligned}
$$

If we combine the three sums and add them up, then we get the following:

$$
\begin{aligned}
&\sum_{1 \le k < j_1} \sum_{y \in S^*(C_1)} p(y) p(b_k) \frac{\sigma_{y b_k}(w)}{\sigma_{y b_k}} + \sum_{j_1 \le k < j_2} \sum_{y \in S^*(C_1)} p(y) p(b_k) \frac{\sigma_{y b_k}(w)}{\sigma_{y b_k}} \\
&+ \sum_{j_2 \le k \le \ell_2} \sum_{y \in S^*(C_1)} p(y) p(b_k) \frac{\sigma_{y b_k}(w)}{\sigma_{y b_k}} \\
&= \sum_{1 \le k < \ell_2} \sum_{y \in S^*(C_1)} p(y) p(b_k) \frac{\sigma_{y b_k}(w)}{\sigma_{y b_k}} \\
&= \sum_{b \in S^*(C_2)} \sum_{y \in S^*(C_1)} p(y) p(b) \frac{\sigma_{y b}(w)}{\sigma_{y b}}
\end{aligned}
\tag{22}
$$

Since we compute this sum for each $C_1, C_2 \in \mathcal{C}$, $C_1 \neq C_2$:

$$\sum_{C_1 \neq C_2 \in \mathcal{C}} \sum_{b \in S^*(C_2)} \sum_{y \in S^*(C_1)} p(y)p(b)\frac{\sigma_{yb}(w)}{\sigma_{yb}} = \sum_{\substack{y \in S^*(C_1), b \in S^*(C_2) \\ C_1, C_2 \in \mathcal{C} \\ C_1 \neq C_2 \\ y \neq w \neq b}} p(y)p(b)\frac{\sigma_{yb}(w)}{\sigma_{yb}} \tag{23}$$

It holds that $y \neq w$ and $w \neq b$ because $w$ is a vertex outside of $C_1$ and $C_2$.

Next, we show the proof for the vertices inside the chain.

### Vertices inside the chain (II)

**Case II.1** We begin with the proof of the computations done in Case II.1. For $b_j$, $1 \leq j \leq j_1$, this is the term:

$$\begin{aligned}
&(\mathrm{t}_{\text{left}}(b_{j_1-1}) - \mathrm{t}_{\text{left}}(b_j)) \cdot \mathrm{t}_{\text{left}}(y_{\ell_1}) \\
&\overset{\text{(i)}}{=} \Big(\sum_{1 \leq k < j_1} p(b_k) - \sum_{1 \leq k \leq j} p(b_k)\Big) \sum_{y \in S^*(C_1)} p(y) \\
&= \sum_{j < k < j_1} \sum_{y \in S^*(C_1)} p(y)p(b_k) \\
&\overset{\text{(ii)}}{=} \sum_{j < k < j_1} \sum_{y \in S^*(C_1)} p(y)p(b_k)\frac{\sigma_{yb_k}(b_j)}{\sigma_{yb_k}} \\
&\overset{\text{(iii)}}{=} \sum_{j < k < j_1} \sum_{y \in S^*(C_1)} p(y)p(b_k)\frac{\sigma_{yb_k}(b_j)}{\sigma_{yb_k}} + \sum_{1 \leq k < j} \sum_{y \in S^*(C_1)} p(y)p(b_k)\frac{\sigma_{yb_k}(b_j)}{\sigma_{yb_k}} \\
&= \sum_{\substack{1 \leq k < j_1 \\ j \neq k}} \sum_{y \in S^*(C_1)} p(y)p(b_k)\frac{\sigma_{yb_k}(b_j)}{\sigma_{yb_k}}
\end{aligned}$$

(i): by definition of $\mathrm{t}_{\text{left}}$ and $\mathrm{t}_{\text{right}}$

(ii): If $j < k < j_1$, $\frac{\sigma_{yb_k}(b_j)}{\sigma_{yb_k}} = 1$, since in Case 1 all shortest paths from $y \in S^*(C_1)$ to $b_k$ enter $C_2$ through $a$ and $b_j$ is between $a$ and $b_k$ (because $j < k < j_1$) and, thus, part of all shortest paths from $y$ to $b_k$.

(iii): If $1 \leq k < j$, $\sigma_{yb_k}(b_j) = 0$, since in Case 1 all shortest paths from $y \in S^*(C_1)$ to $b_k$ enter $C_2$ through $a$ and $b_j$ is not between $a$ and $b_k$ (because $k < j$).

**Case II.2 (II.3 is analogous)** For Case II.2, depending on the index $j$, $1 \leq j \leq \ell_2$, we have to do different computations for $b_j$. But before, we define

$$f(j) := \begin{cases} \sum_{j_1 \leq k < j_2} p(y^k_{\text{mid}})p(b_k)\frac{\sigma_{y^k_{\text{mid}} b_k}(b_j)}{y^k_{\text{mid}} b_k}, \text{ if } y^j_{\text{mid}} \text{ exist} \\ 0, \text{ otherwise} \end{cases},$$

because again, $y_{\text{mid}}^j$ may or may not exist and thus, its contribution to the betweenness centrality of the other vertices may be zero.

**Case II.2.1** If $1 \leq j < j_1$, then we have to compute term (18):

$$\begin{aligned}
&\sum_{j_1 \leq k < j_2} \mathrm{t}_{\text{left}}(y_{\text{left}}^k) p(b_k) + f(j) \\
&\overset{\text{(i)}}{=} \sum_{j_1 \leq k_1 < j_2} \sum_{1 \leq k_2 \leq I_{\text{left}}^{k_1}} p(y_{k_2}) p(b_{k_1}) + f(j) \\
&\overset{\substack{\text{analog} \\ \text{to (ii)}}}{=} \sum_{j_1 \leq k_1 < j_2} \sum_{1 \leq k_2 \leq I_{\text{left}}^{k_1}} p(y_{k_2}) p(b_{k_1}) \frac{\sigma_{y_{k_2} b_{k_1}}(b_j)}{\sigma_{y_{k_2} b_{k_1}}} + f(j) \\
&\overset{\substack{\text{analog} \\ \text{to (iii)}}}{=} \sum_{j_1 \leq k_1 < j_2} \Big( \sum_{1 \leq k_2 \leq I_{\text{left}}^{k_1}} p(y_{k_2}) p(b_{k_1}) \frac{\sigma_{y_{k_2} b_{k_1}}(b_j)}{\sigma_{y_{k_2} b_{k_1}}} \\
&+ \sum_{I_{\text{right}}^{k_1} \leq k_2 \leq \ell_1} p(y_{k_2}) p(b_{k_1}) \frac{\sigma_{y_{k_2} b_{k_1}}(b_j)}{\sigma_{y_{k_2} b_{k_1}}} \Big) + f(j) \\
&= \sum_{j_1 \leq k_1 < j_2} \sum_{1 \leq k_2 \leq \ell_2} p(y_{k_2}) p(b_{k_1}) \frac{\sigma_{y_{k_2} b_{k_1}}(b_j)}{\sigma_{y_{k_2} b_{k_1}}} \\
&= \sum_{j_1 \leq k_1 < j_2} \sum_{y \in S^*(C_1)} p(y) p(b_{k_1}) \frac{\sigma_{y b_{k_1}}(b_j)}{\sigma_{y b_{k_1}}}
\end{aligned}$$

**Case II.2.2** If $j_2 \leq j \leq \ell_2$, then we have to compute term (19):

$$\begin{aligned}
&\sum_{j_1 \leq k < j_2} \mathrm{t}_{\text{right}}(y_{\text{right}}^k) p(b_k) + f(j) \\
\overset{\substack{\text{analog to} \\ \text{Case II.2.1}}}{=} &\sum_{j_1 \leq k_1 < j_2} \sum_{y \in S^*(C_1)} p(y) p(b_{k_1}) \frac{\sigma_{y b_{k_1}}(b_j)}{\sigma_{y b_{k_1}}}
\end{aligned}$$

**Case II.2.3** If $j_1 \leq j < j_2$, then we have to compute term (20):

$$\sum_{j<k<j_2} \mathrm{t}_{\mathrm{left}}(y^k_{\mathrm{left}})p(b_k) + \sum_{j_1 \leq k < j} \mathrm{t}_{\mathrm{right}}(y^k_{\mathrm{right}})p(b_k) + f(j)$$

$$\overset{\substack{\text{(i),}\\ \text{analog}\\ \text{to (ii)}}}{=} \sum_{j<k_1<j_2} \sum_{1 \leq k_2 \leq I^{k_1}_{\mathrm{left}}} p(y_{k_2})p(b_{k_1}) \frac{\sigma_{y_{k_2}b_{k_1}}(b_j)}{\sigma_{y_{k_2}b_{k_1}}}$$

$$+ \sum_{j_1 \leq k < j} \sum_{I^{k_1}_{\mathrm{right}} \leq k_2 \leq \ell_1} p(y_{k_2})p(b_{k_1}) \frac{\sigma_{y_{k_2}b_{k_1}}(b_j)}{\sigma_{y_{k_2}b_{k_1}}} + f(j)$$

$$\overset{\text{(iv)}}{=} \sum_{j<k_1<j_2} ($$

$$\sum_{1 \leq k_2 \leq I^{k_1}_{\mathrm{left}}} p(y_{k_2})p(b_{k_1}) \frac{\sigma_{y_{k_2}b_{k_1}}(b_j)}{\sigma_{y_{k_2}b_{k_1}}} + \sum_{I^{k_1}_{\mathrm{right}} \leq k_2 \leq \ell_1} p(y_{k_2})p(b_{k_1}) \frac{\sigma_{y_{k_2}b_{k_1}}(b_j)}{\sigma_{y_{k_2}b_{k_1}}})$$

$$+ \sum_{j_1 \leq k_1 < j} ($$

$$\sum_{I^{k_1}_{\mathrm{right}} \leq k_2 \leq \ell_1} p(y_{k_2})p(b_{k_1}) \frac{\sigma_{y_{k_2}b_{k_1}}(b_j)}{\sigma_{y_{k_2}b_{k_1}}} + \sum_{1 \leq k_2 \leq I^{k_1}_{\mathrm{left}}} p(y_{k_2})p(b_{k_1}) \frac{\sigma_{y_{k_2}b_{k_1}}(b_j)}{\sigma_{y_{k_2}b_{k_1}}}) + f(j)$$

$$= \sum_{j<k<j_2} ( \sum_{y \in S^*(C_1)} p(y)p(b_k) \frac{\sigma_{yb_k}(b_j)}{\sigma_{yb_k}}) + \sum_{j_1 \leq k < j} ( \sum_{y \in S^*(C_1)} p(y)p(b_k) \frac{\sigma_{yb_k}(b_j)}{\sigma_{yb_k}})$$

$$= \sum_{\substack{j_1 \leq k < j_2 \\ j \neq k}} \sum_{y \in S^*(C_1)} p(y)p(b_k) \frac{\sigma_{yb_k}(b_j)}{\sigma_{yb_k}}$$

(iv): If $j < k_1 < j_2$ and $I^{k_1}_{\mathrm{right}} \leq k_2 \leq \ell_1$, then $\sigma_{y_{k_2}b_{k_1}}(b_j) = 0$ since all shortest paths from $y_{k_2}$ to $b_{k_1}$ enter $C_2$ through $z$ and $b_j$ is not between $z$ and $b_{k_1}$ (because $j < k_1$).
If $j_1 \leq k_1 < j$ and $1 \leq k_2 \leq I^{k_1}_{\mathrm{left}}$, then $\sigma_{y_{k_2}b_{k_1}}(b_j) = 0$ since all shortest paths from $y_{k_2}$ to $b_{k_1}$ enter $C_2$ through $a$ and $b_j$ is not between $a$ and $b_{k_1}$ (because $k_1 < j$).

Independent of the index $j$ in this case, we get the correct result.

**Case II.4** The proof of Case II.4 is analogous to the proof of Case II.1. We have to compute the term:

$$\mathrm{t}_{\mathrm{right}}(b_{j_2}) - \mathrm{t}_{\mathrm{right}}(b_j)) \cdot \mathrm{t}_{\mathrm{left}}(y_{\ell_1})$$

$$\overset{\substack{\text{analog to}\\ \text{Case II.1}}}{=} \sum_{\substack{j_2 \leq k < \ell_2 \\ j \neq k}} \sum_{y \in S^*(C_1)} p(y)p(b_k) \frac{\sigma_{yb_k}(b_j)}{\sigma_{yb_k}}$$

Combining all cases and doing this for all $C_1, C_2 \in \mathcal{C}$, $C_1 \neq C_2$, is already shown in (22) and (23).

**Running time** The running time of Step 2 depends on the cases that apply for each pair of chains $C_1$, $C_2$. In the best case, only Case 1 or Case 4 apply for the vertices in $C_2$. In Case 1 or Case 4, the computations for each pair of chains are linear in time, since we have to increase the betweenness centrality of all vertices which lie on some shortest path by a constant term and the number of these vertices is dependent from $n$. This yields a total running of

$$(\sum_{\substack{C_1,C_2\in\mathcal{C}\\C_1\neq C_2}} O(1))\cdot O(n) = O(|\mathcal{C}|^2\cdot n) = O(k^2\cdot n)$$

for Step 2. Recall that we can bound the number of chains in the graph the feedback edge number $k$. In the worst case though, only Case 2 or Case 3 apply. In those cases we have to compute a sum, whose size depends on the length of $C_2$, rather than a just a constant term for each pair $C_1$, $C_2$ of chains. Since the length of a chain depends on $n$, this results in a total running time of

$$(\sum_{\substack{C_1,C_2\in\mathcal{C}\\C_1\neq C_2}} O(n))\cdot O(n) = O(k^2\cdot n^2)$$

for Step 2. □

## 5.4 Step 3: Shortest paths between vertices of a single chain

We now look at each chain separately and consider every shortest path between each pair of vertices in that chain. The respective sum that we now calculate for every $v \in V$ is the following (see (11)):

$$\sum_{\substack{s,t\in S^*(C)\\C\in\mathcal{C}\\s\neq v\neq t}} p(s)\cdot p(t)\frac{\sigma_{st}(v)}{\sigma_{st}}.$$

In Algorithm 1, this is displayed in line 29.

Let $C = (x, y_1, ..., y_\ell, z)$ be any chain. We distinguish the two cases, whether the shortest paths from $y_0$ to $y_\ell$ go through $C$ or not:

**Case 1:** $\ell - 1 < d(x,z) + 2$ In this case, we can look at $C$ as a path from $y_1$ to $y_\ell$. For paths, Unnithan et al. [Unn+14] showed that the betweenness centrality of each vertex in that path is computed by multiplying the number of vertices left to that vertex by the number of vertices right to that vertex. Since we also have to take into account the vertices originally connected to the vertices in the chain, for each $y_i$, $2 \leq i \leq \ell - 1$, we have to increase its betweenness centrality by

$$\mathrm{t}_{\mathrm{left}}(i-1) \cdot \mathrm{t}_{\mathrm{right}}(i+1).$$

No shortest path between the vertices in $C$ leaves the chain. Hence, we do not have to increase the betweenness centrality of vertices outside the chain.

**Case 2:** $\ell - 1 \geq d(x,z) + 2 = d(y_1, y_\ell)$ Since the distance from $y_1$ to $y_\ell$ is smaller than (or equal to) $\ell$ (the length of the chain), there may be shortest paths from $y_1$ to $y_\ell$ which leave the chain and go through the vertices $x$ and $z$. Then, the betweenness centrality of vertices outside the chain is also affected by the shortest paths between the vertices in the chain. Formally, for a chain $C = (x, y_1, ..., y_\ell, z)$, the shortest path between $y_i$ and $y_j$, $1 \leq i < j \leq \ell$, stays inside $C$, if the following inequality holds:

$$j - i < i + \ell - j + 1 + d_G(x,z) \tag{24}$$

In the inequality, the term $(j-i)$ on the left-hand-side is the length of the path $(y_i, ..., y_j)$, which is the path from $y_i$ to $y_j$ that stays inside $C$. The term $(i + \ell - j + 1 + d_G(x,z))$ on the right-hand-side is the length of the path $(y_i, ..., y_1, x, ..., z, y_\ell, ...y_j)$, which is the path from $y_i$ to $y_j$ that leaves $C$ and passes trough $x$ and $z$. Note that in this case there is exactly one shortest path between $y_i$ and $y_j$. If the inequality is not true though, then we can distinguish two cases: If

$$j - i > i + \ell - j + 1 + d_G(x,z) \tag{25}$$

holds, then the shortest paths from $y_i$ to $y_j$ pass through $x$ and $z$. In this case, there are $\sigma_{xz}$ shortest paths from $y_i$ to $y_j$. If

$$j - i = i + \ell - j + 1 + d_G(x,z) \tag{26}$$

is true, then the path inside $C$ and the path outside $C$ have the same length and, thus, are both shortest paths. This results in a total of $\sigma_{xz} + 1$ shortest paths between $y_i$ and $y_j$.

We will use the above inequalities to construct formulas that compute the betweenness centrality for all $y \in S^*(C)$ lying on some shortest paths between the vertices inside $C$. Since we want a linear computing time per chain, we use the same approach as in Section 4: We start by constructing a sum that computes the betweenness centrality of $y_1$ in linear time. Next, we iteratively compute the betweenness centrality for all other $y_i \in S^*(C)$ by subtracting the amount of betweenness that is given for shortest paths that pass through $y_{i-1}$ but do not pass through $y_i$ and adding the amount of betweenness that is given for shortest paths that do not pass through $y_{i-1}$ but pass through $y_i$. Each step of the iteration needs constant time. This way we achieve a total computing time of $O(\ell)$ per chain.

After doing so, we still have to handle the vertices lying on some shortest paths between vertices of $C$ but are not part of $C$ itself. We do this by constructing a sum, which represents the amount shortest paths that leave $C$. For each $w \in V \setminus S^*(C)$, we have to increase its betweenness centrality by a value depending on the constructed sum. This part has a running time of $O(\ell + n) = O(n)$ per chain.

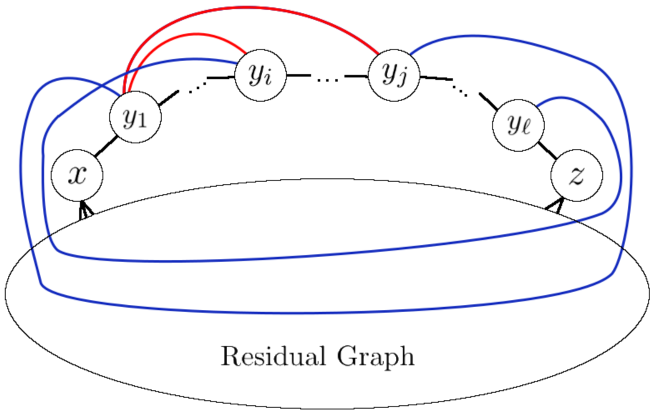



Figure 8: A single chain with both shortest paths staying in the chain (red line) and leaving the chain (blue line)

**Computations for $y_1$** The vertex $y_1$ can only be part of those shortest paths between $y_i$ and $y_j$, $1 < i < j \leq \ell$, that leave $C$, because, otherwise, $y_1$ had to lie between $y_i$ and $y_j$, which is impossible. See Figure 8 for the chain $C$. The red lines between the vertices $y_1$ and $y_i$ and between $y_1$ and $y_j$ represent that between those pairs of vertices there is a shortest path that stays in $C$. The blue lines between the vertices $y_1$ and $y_j$ and between $y_i$ and $y_\ell$ indicate that between those pairs of vertices there is a shortest path leaving the chain. We will create a sum that for each $i$, adds up the number of shortest paths to each $y_j$ leaving $C$. In Figure 8 one such shortest paths is represented by the blue line between $y_i$ and $y_\ell$. The shortest paths from $y_i$ to $y_j$ leave $C$, if and only if inequality (25) holds. We want to find the smallest $j$, such that the shortest paths between $y_i$ and $y_j$ still leave $C$. This is achieved by solving inequality (25) for $j$:

$$\begin{aligned} & j - i > i + \ell - j + 1 + d_G(x, z) \\ \iff \quad & j > \frac{\ell + 1 + d_G(x, z)}{2} + i \end{aligned} \tag{27}$$

For easier reading we define

$$\gamma(i) := \frac{\ell + 1 + d_G(x, z)}{2} + i. \tag{28}$$

Now, we want to find the biggest $i$ such that there still is a $y_j$, $i < j \leq \ell$, such that there is a shortest path between $y_i$ and $y_j$ leaving $C$. Therefore, we again have to solve inequality (25). This time, we solve it for $i$ and set $j = \ell$, since we need the biggest possible $y_j$:

$$\begin{aligned} & \ell - i > i + \ell - \ell + 1 + d_G(x, z) \\ \iff \quad & i < \frac{\ell - d_G(x, z) - 1}{2} \end{aligned}$$

Combining the results of both inequalities results that for each $i$, $2 \leq i < \frac{\ell - d_G(x,z)-1}{2}$, there is a shortest path to each $y_j$, $\gamma(i) < j \leq \ell$. If we create a sum over all possible pairs $y_i$, $y_j$ (and also taking into account the vertices originally connected to the graph), for the betweenness centrality of $y_1$ in $C$, we get:

$$\sum_{i=2}^{\lfloor \frac{\ell - d_G(x,z)-1}{2} \rfloor} p(y_i)(\sum_{j=\lceil \gamma(i) \rceil}^{\ell} p(y_j)) = \sum_{i=2}^{\lfloor \frac{\ell - d_G(x,z)-1}{2} \rfloor} p(y_i)\, \mathrm{t}_{\mathrm{right}}(y_{\lceil \gamma(i) \rceil})$$

But the above sum only considers these pairs of vertices $y_i$ and $y_j$, $1 \leq i < j \leq \ell$, for which *all* shortest path between them leave $C$. We still need to consider those pairs, where there is a shortest path between them staying in $C$ and some shortest paths leaving $C$. In Figure 8 this is the case between the vertices $y_1$ and $y_j$ since there is both a red line and a blue line connecting $y_1$ and $y_j$. Those pairs do not necessarily exist: Whether they exist or not depends on the length of the chain, $\ell$, and the distance between $x$ and $z$, $d_G(x,z)$, as we will see later (in (29)). For now, assume those pairs exist. We need to look at equation (26) again. For each $i$, there can exist only one $j$, such that between $y_i$ and $y_j$ there are both shortest paths leaving $C$ and staying in $C$. We determine that $j$ by solving equation (26) for $j$:

$$\begin{aligned} & j - i = i + \ell - j + 1 + d_G(x,z) \\ \iff \quad & j = \frac{\ell + 1 + d_G(x,z)}{2} + i \end{aligned}$$

This is the same result as in inequality (27) and already defined in (28). So, between $y_i$ and $y_{\gamma(i)}$, if $\gamma(i)$ is an integer and $y_{\gamma(i)}$ exists, there is exactly one shortest path staying in $C$ and there are $\sigma_{xz}$ shortest paths leaving $C$. The vertex $y_1$ lies on all the shortest paths leaving $C$ but not on that shortest path staying in $C$. This results in a value of $\frac{\sigma_{xz}}{\sigma_{xz}+1}$ for each pair $y_i$ and $y_{\gamma(i)}$ to add to the betweenness centrality of $y_1$.

Combining all results, for the betweenness centrality of $y_1$ in $C$, $C_B^C(y_1)$, we finally get:

$$C_B^C(y_1) \coloneqq \sum_{i=2}^{\lfloor \frac{\ell - d_G(x,z)-1}{2} \rfloor} p(y_i)(\mathrm{t}_{\mathrm{right}}(y_{\lceil \gamma(i) \rceil + 1}) + p(\lceil \gamma(i) \rceil)\omega(C)),$$

where

$$\omega(C = (x, y_1, ..., y_\ell, z)) = \begin{cases} 1, & \text{if } \frac{\ell + 1 + d_G(x,z)}{2} \text{ is odd} \\ \frac{\sigma_{xz}}{\sigma_{xz}+1}, & \text{otherwise} \end{cases} \tag{29}$$

represents the fact that there may exist pairs of vertices, where there are both shortest paths staying in $C$ and leaving $C$ between them: If they they exist, then we need the factor of $\frac{\sigma_{xz}}{\sigma_{xz}+1}$, if not, then we need the factor 1.

We will continue with the computations that need to be done for all further $y_i \in C$, $1 < i \leq \ell$.

**Computations for $y_i$, $1 < i \leq \ell$** In this part, we compute the betweenness centrality for the remaining vertices in the chain. As already mentioned, we do this iteratively: For $y_i$, $1 < i \leq \ell_2$, we first take the result of its predecessor, $C_B^C(y_{i-1})$. Next, we subtract the amount of all shortest paths, that start in $y_i$, since, if $y_i$ is an endpoint of a shortest path, then this path must not be considered for the betweenness centrality of $y_i$. Then, we have to add all shortest paths, that start in $y_{i-1}$ and pass through $y_i$. The shortest paths from $y_{i-1}$ were not considered in the computation of $C_B^C(y_{i-1})$, since $y_{i-1}$ is an endpoint of them. The formula is similar to the one for balloons:

$$C_B^C(y_i) = C_B^C(y_{i-1}) - p(y_i)\alpha(i) + p(y_{i-1})\beta(i)$$

The term $\alpha(i)$ represents the number of shortest paths that need to be subtracted whereas the term $\beta(i-1)$ is the number of shortest paths that need to be added.

We will first show how to calculate $\alpha(i)$. We need to subtract all shortest paths that start in $y_i$ and pass through $y_{i-1}$. These shortest paths can again either stay in $C$ or leave $C$. For both cases, we need to solve an inequality. We first have to find the smallest possible $i'$, $0 < i' < i-1$, such that the shortest path from $y_{i'}$ to $y_i$ stays inside $C$. Therefore, we solve (24) for $i'$:

$$\begin{aligned} & i - i' < i' + \ell - i + 1 + d_G(x,z) \\ \iff & i' > i - \frac{l + 1 + d_G(x,z)}{2} \end{aligned}$$

Next, we need the smallest possible $i'$, $i < i' \leq \ell$, such that the shortest path from $y_i$ to $y_{i'}$ leaves $C$. Therefore, we solve (25) for $i'$:

$$\begin{aligned} & i' - i > i + \ell - i' + 1 + d_G(x,z) \\ \iff & i' > i + \frac{l + 1 + d_G(x,z)}{2} \end{aligned}$$

For easier reading, we define:

$$\delta := \frac{l + 1 + d_G(x,z)}{2}$$

If we combine both results, we get:

$$\begin{aligned} \alpha(i) := & \, \mathrm{t}_{\mathrm{right}}(i - \lceil\delta\rceil + 1) - \mathrm{t}_{\mathrm{right}}(i-1) + \mathrm{t}_{\mathrm{right}}(i + \lceil\delta\rceil + 1) \\ & + \omega'(C)p(i - \lceil\delta\rceil) + \omega(C)p(i + \lceil\delta\rceil) \end{aligned},$$

where

$$\omega'(C = (x, y_1, ..., y_\ell, z)) = \begin{cases} 1 & \text{, if } \frac{\ell+1+d_G(x,z)}{2} \text{ is odd} \\ \frac{1}{\sigma_{xz}+1} & \text{, otherwise} \end{cases}. \tag{30}$$

Note the factors $\omega(C)$ and $\omega'(C)$ in the formula. These factors are needed, because if between two vertices $y_i$ and $y_j$, there is both one shortest path that stays inside $C$ and a shortest path that leaves $C$, then the vertices lying on one of these paths do not lie on all shortest paths between $y_i$ and $y_j$. Each vertex can only lie on either the single shortest path that stays in $C$ or on each shortest path that leaves $C$. This results in a factor of $\frac{1}{\sigma_{xz}+1}$ or $\frac{\sigma_{xz}}{\sigma_{xz}+1}$, respectively.

Now, for $\beta$, we have to find similar indices. We need to add up all shortest paths, that start in $y_{i-1}$ and pass through $y_i$. Therefore, we need to find the biggest possible $i'$, $i < i' \leq \ell$, such that the shortest path from $y_{i-1}$ to $y_{i'}$ stays in $C$ and we have to find the biggest possible $i'$, $0 < i' < i-1$, such that the shortest paths from $y_{i'}$ to $y_{i-1}$ leave $C$. Again, we have to solve two inequalities to get these indices. Since this is analogous as for $\alpha$ ($i$ is just replaced by $(i-1)$), we only give the results:

$$\begin{aligned}\beta(i) \coloneqq\, &\mathrm{t}_{\mathrm{left}}((i-1)+\lfloor\delta\rfloor-1) - \mathrm{t}_{\mathrm{left}}(i) + \mathrm{t}_{\mathrm{left}}((i-1)-\lfloor\delta\rfloor-1)\\ &+ \omega'(C)p((i-1)+\lfloor\delta\rfloor) + \omega(C)p((i-1)-\lfloor\delta\rfloor).\end{aligned}$$

**Vertices outside the chain** In both computations, we only treated the vertices inside $C$. But since some shortest paths leave the chain, we need to also look at the vertices lying on such shortest paths. Therefore, we sum up all shortest paths that leave $C$. We already did this for the computation of $C_B^C(y_1)$ (see (5.4)). But we need to also include all shortest paths starting in $y_1$ itself. These are not included in $C_B^C(y_1)$, because $y_1$ obviously is an endpoint of them. Thus, the sum begins with $i = 1$. For $w \in V \setminus S^*(C)$, we have to compute the following sum. We have to add the factor $\frac{\sigma_{xz}(w)}{\sigma_{xz}}$ since each $w$ does not necessarily lie on all shortest paths between two vertices in the chain.

$$\begin{aligned}C_B^C(w) &= \sum_{i=1}^{\lfloor\frac{\ell-d_G(x,z)-1}{2}\rfloor} p(y_i)((\mathrm{t}_{\mathrm{right}}(y_{\lceil\gamma(i)\rceil+1}))\frac{\sigma_{xz}(w)}{\sigma_{xz}} + p(\lceil\gamma(i)\rceil)\omega''(C)\sigma_{xz}(w))\\ &= \sigma_{xz}(w)\sum_{i=1}^{\lfloor\frac{\ell-d_G(x,z)-1}{2}\rfloor} p(y_i)((\mathrm{t}_{\mathrm{right}}(y_{\lceil\gamma(i)\rceil+1})\frac{1}{\sigma_{xz}} + p(\lceil\gamma(i)\rceil)\omega''(C))\end{aligned},$$

where

$$\omega''(C = (x, y_1, ..., y_\ell, z), w) = \begin{cases}\frac{1}{\sigma_{xz}} & \text{, if } \frac{\ell+1+d_G(x,z)}{2} \text{ is odd}\\ \frac{1}{\sigma_{xz}+1} & \text{, otherwise}\end{cases}.$$

The sum does not depend on the vertex $w$. Thus, we only have to compute the sum once and then multiply it with the term $\sigma_{xz}(w)$. This ensures a linear running time per chain. We define $\omega''$, since between some pairs of vertices in the chain, there may exist both shortest paths leaving $C$ and staying in $C$. Each $w$ only lies on the $\sigma_{xz}(w)$ shortest paths that leave $C$ and cannot lie on the shortest path that stays in $C$. This results in a factor of $\frac{\sigma_{xz}(w)}{\sigma_{xz}+1}$ if those pairs exist, or in a factor of $\frac{\sigma_{xz}(w)}{\sigma_{xz}}$ if not. The correctness of the above formulas follows from their construction.

**Running time** The computation of $C_B^C(y_1)$ is linear, depending on $\ell$, the length of the chain. The computation of all other $C_B^C(y_i)$, $1 < i \leq \ell$, is constant. Hence, for the vertices inside $C$, we have a running time of $O(\ell)$.

The computation of $C_B^C(w)$, $w \in V \setminus S^*(C)$ is linear in time for the first $w$, again depending on $\ell$ and constant for all following $w$. Since the size of $|V \setminus S^*(C)|$ is bounded by $n$, we have a running time of $O(\ell + n)$ per chain. In total, the computing time for a single chain is $O(\ell) + O(\ell + n) = O(n)$ and $O(|\mathcal{C}| \cdot n) = O(k \cdot n)$ for all chains.

### 5.5 Final running time

We now summarize the results of Section 4 and Section 5 and thus proof the correctness of Theorem 1.1. The correctness of the computations we did in both sections were already proven in the respective section. Hence, we yet need to show that the final running time of our algorithm is $O(k^2 \cdot n^2)$: Therefore, look at the running times of the computations done in Section 4 and in the subsections of Section 5:

- The running time of the computations done in Section 4 is $O(n)$.
- The running times of the computations done in Section 5 are:
  - $O(k^2 \cdot n)$ for Section 5.2
  - $O(k^2 \cdot n^2)$ for Section 5.3
  - $O(k \cdot n)$ for Section 5.4

Adding up all running times, we get:

$$O(n) + O(k^2 \cdot n) + O(k^2 \cdot n^2) + O(k \cdot n) = O(k^2 \cdot n^2).$$

In the following section we give a conclusion summarizing the ideas and results of this work as well as an approach to further improve our algorithm.

## 6 Conclusion & Outlook

Baglioni et al. [Bag+12] showed in their work that when it comes to computing the betweenness centrality of each vertex in a graph it makes sense to treat degree-one vertices separately and exploit their unique characteristics. They can be removed from the input graph in linear time, and with little adjustments in Brandes' algorithm, do not play any role later, when actually performing Brandes' algorithm on the residual graph. This work is an attempt to show that it makes sense to treat degree-two vertices separately, too. In this work we could not delete all vertices of degree two but only some of them. If we were able to delete all vertices of degree two from the graph we could define a kernel on the input graph with only vertices of degree at least three left. By making use of the feedback edge number $k$ we can bound the size of the kernel and can provide an algorithm being linear in $n$ and $m$ and only polynomial in $k$. For small $k$ this can be faster than Brandes' algorithm.

The algorithm that we presented in this work however, only has a running time of $O(k^2 \cdot n^2)$, which is not linear in $n$ and thus, not necessarily faster than the algorithm of Brandes, which runs in $O(n \cdot m)$. We conjecture that with little improvements a running time of $O(k^2 \cdot (n+m))$ can be achieved. The needed improvement is mentioned in the next paragraph. Additionally, our chosen parameter $k$, the feedback edge number, can become relatively big for big instances of networks. This is not desirable and motivates to find better parameters to bound the running time of such algorithms.

**Improvement** The factor $n^2$ results from the need of computing sums dependent on $n$ in Cases 2 and 3 in Step 2 (see Section 5.3). Replacing these sums by constant terms would yield the desired running time of $O(k^2 \cdot n)$ for Step 2. We are confident that this is indeed possible. The effective speed up in a practical environment needs then to be evaluated in experiments. At last follows an approach to define a problem kernel by deleting *all* vertices of degree two from the graph.

**Towards a problem kernel** We only deleted those vertices of degree two that were part of a cyclic structure. Here, we want to give an approach on how to delete other degree-two vertices from the input graph. This would speed up the execution of Brandes' (modified) algorithm (see line 17 in Algorithm 1). Since we delete all vertices of degree two, we would also obtain a problem kernel of size $O(k)$ (see [Mer+17, proof of Thm 2.3]).

Our approach is to replace each chain with a weighted edge before running Brandes' algorithm on the graph. The weight of the new edge is set to the length of the chain. This way, we do not alter the lengths of the shortest paths between the remaining vertices. Of course, this transforms the input graph into an edge-weighted graph. But because Brandes' algorithm also works on edge-weighted graphs [Bra01], this is not a problem. When replacing the chains with edges though, some new problems arise. For example, when running Brandes' algorithm on the residual graph with only vertices with a degree of at least three left, we do not increase the betweenness centrality of the just deleted degree-two vertices. This has to be done separately afterwards. In this work we were not able to overcome all such technical details. This needs to be done in future work. Future work may also exploit other parameters than the feedback edge number to develop faster and more feasible algorithms for calculating the betweenness centrality.

## Literature


[Bag+12] M. Baglioni, F. Geraci, M. Pellegrini, and E. Lastres. "Fast exact computation of betweenness centrality in social networks". In: *Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012).* IEEE Computer Society. 2012, pp. 450–456 (cit. on pp. 5, 6, 8, 9, 12, 13, 20, 48).

[Bav48] A. Bavelas. "A mathematical model for group structures". In: *Human organization* 7.3 (1948), pp. 16–30 (cit. on p. 5).

[Bra01] U. Brandes. "A faster algorithm for betweenness centrality". In: *Journal of mathematical sociology* 25.2 (2001), pp. 163–177 (cit. on pp. 5, 6, 8, 49).

[For10] S. Fortunato. “Community detection in graphs”. In: *Physics reports* 486.3 (2010), pp. 75–174 (cit. on p. 6).

[Fre77] L. C. Freeman. “A set of measures of centrality based on betweenness”. In: *Sociometry* (1977), pp. 35–41 (cit. on p. 5).

[Gia+17] A. C. Giannopoulou, G. B. Mertzios, and R. Niedermeier. “Polynomial fixed-parameter algorithms: A case study for longest path on interval graphs”. In: *Theoretical Computer Science* (2017) (cit. on p. 7).

[Mer+17] G. B. Mertzios, A. Nichterlein, and R. Niedermeier. “The power of data reduction for matching.” In: (2017) (cit. on pp. 9, 49).

[NG04] M. E. Newman and M. Girvan. “Finding and evaluating community structure in networks”. In: *Physical review E* 69.2 (2004), p. 026113 (cit. on p. 6).

[RS10] M. Rubinov and O. Sporns. “Complex network measures of brain connectivity: uses and interpretations”. In: *Neuroimage* 52.3 (2010), pp. 1059–1069 (cit. on p. 6).

[Unn+14] R. Unnithan, S. Kumar, B. Kannan, and M. Jathavedan. “Betweenness centrality in Some classes of graphs”. In: *International Journal of Combinatorics* 2014 (2014) (cit. on p. 42).