

UPDATING KATZ CENTRALITY BY COUNTING WALKS*

FRANCESCA ARRIGO[†], DANIELE BERTACCINI[‡], AND ALESSANDRO FILIPPO[‡]

Abstract. We develop efficient and effective strategies for the update of Katz centralities after node and edge removal in simple graphs. We provide explicit formulas for the “loss of walks” a network suffers when nodes/edges are removed, and use these to inform our algorithms. The theory builds on the newly introduced concept of \mathcal{F} -avoiding first-passage walks. Further, bounds on the change of total network communicability are also derived. Extensive numerical experiments on synthetic and real-world networks complement our theoretical results.

Key words. Complex network, matrix function, combinatorics, centrality measure, Katz centrality.

MSC codes. 05C50, 05C90, 68R10, 65F99

1. Motivation. Katz centrality is one of the most popular centrality indices in network science [18, 29, 34], which has successfully been applied to a wide range of problems, such as, for example, the identification of potential disease genes in protein-interaction networks [42] or the optimization of artificial neural networks [31]. This measure’s popularity is due to its interpretability in terms of walks around a graph but also to the easiness of its computation [5, 35]. Indeed, Katz centrality can be expressed as the solution to a (usually sparse) linear system, an extensively studied problem that is accompanied by a well-developed set of numerical linear algebra software libraries; see, e.g., [9, 15, 37] and references therein. Moreover, Katz centrality can be easily extended to the setting of dynamic networks [22] or specialised to only consider particular types of walks [21] without affecting its easiness of computation, thus making it possibly the most widely used walk-based centrality measure in the field.

In this paper, we consider the problem of approximately updating the vector of Katz centrality and its “personalised” counterpart [7] after the sequential removal/failure of a set of nodes or edges from a simple network.

This is particularly relevant in real life applications. Indeed, in many real-world networks nodes and/or edges evolve over time [1, 13, 17]. For example, an edge being removed could correspond to road closure within a road network, while a node being deleted could model a user leaving a social media platform. When centrality is of interest in these and similar situations, recomputing the vector of centrality scores may not always be feasible, especially for extremely large networks. Approximating strategies like the ones presented in this paper may be devised instead. Additionally, our results may also be useful in the context of network design and network optimization, where simulating the removal of edges is often used as a way to address issues related to the robustness of the network or the presence communication

*Submitted to the editors DATE.

Funding: The work of DB was partially supported by MIUR Excellence Department Project MatMod@TOV 2023-2027 awarded to the Department of Mathematics, University of Rome Tor Vergata and by the Horizon 2020 Project Energy oriented Centre of Excellence: toward exascale for energy (EoCoE II). DB and AF are members of the Gruppo Nazionale Calcolo Scientifico-Istituto Nazionale di Alta Matematica (GNCS-INdAM)”

[†]Department of Mathematics and Statistics, University of Strathclyde, Glasgow, UK (francesca.arrigo@strath.ac.uk).

[‡]Department of Mathematics, University of Rome Tor Vergata, Rome, Italy (bertaccini@mat.uniroma2.it, filippo@mat.uniroma2.it).

bottlenecks [2, 11, 25, 27].

Several updating strategies for centrality vectors have been devised over the years [30, 38] and, for walk-based centralities, the problem has been extensively studied (whether directly or indirectly) from the perspective of small rank updates of matrix functions and sensitivity of their entries; see, e.g., [4, 8, 16, 36, 39, 40]. For Katz centrality in particular, a few ad-hoc strategies have been proposed. In [32] the authors proposed to update centralities via an iterative refinement method applied to the original Katz vector, while [33] developed a visiting algorithm to modify the overall count of walks up to a specific length.

We take a different approach and devise an approximation strategy that builds on the combinatorics of walks that are lost due to node/edge removal and, more specifically, on recurrence formulas that count *first-passage walks* [28], i.e., walks between two nodes that exclusively visit the final node in their last step.

From our approximations, we also derive bounds on the change in the total network communicability induced by Katz centrality [6] when edges/nodes are removed from the original graph. These bounds, together with our formulas for the updated Katz centrality scores, could be exploited to identify the new information propagators in social networks after node/edge removal; see [41].

1.1. Background and notation. In the following, we briefly recall some standard definitions from network science and set the notation that will be used throughout. We refer the interested reader to [18, 34] for more details.

An unweighted *graph* or *network* $G = (V, E)$ consists of a set of *nodes* (*vertices*) $V = \{1, 2, \dots, n\}$ and a set $E \subset V \times V$ of *edges* (*links*) between them. If E is symmetric, then G is said to be *undirected*; *directed* otherwise. An undirected graph with no repeated edges nor loops, i.e., edges connecting a node to itself, is said to be *simple*. Henceforth, unless otherwise specified, all graphs considered are assumed to be simple.

Two nodes $i, j \in V$ are said to be *adjacent* if there exists an edge between them. We will denote such edge as $\{i, j\}$ or $i \sim j$. We remark that every undirected edge $i \sim j$ can equivalently be viewed as a pair of opposite directed edges $i \rightarrow j$ and $j \rightarrow i$. The *degree* of node i , denoted by $\deg(i)$, is the number of nodes adjacent to i . The *adjacency matrix* $A \in \mathbb{R}^{n \times n}$ of G is entrywise defined as

$$A_{ij} = \begin{cases} 1 & \text{if } i \sim j \\ 0 & \text{otherwise.} \end{cases}$$

It follows from this definition, that $A^T = A$ whenever the graph is simple.

A *walk of length r* from node i to node j is an ordered sequence of $r + 1$ nodes $i_0 = i, i_1, \dots, i_r = j$ such that $\{i_k, i_{k+1}\} \in E$ for all $k = 0, 1, \dots, r - 1$. A walk is said to be *closed* if $i_0 = i_r$.

A *path* is a walk with no repeated vertices. We will say that a walk i_0, i_1, \dots, i_r of length r *passes through* or *visits* node w if $i_k = w$ for some $k = 0, \dots, r$. Similarly, we say that a walk *visits edge* $e = \{u, v\}$ if it passes through nodes u, v either as $i_k = u$ and $i_{k+1} = v$ or as $i_k = v$ and $i_{k+1} = u$, for some $k = 0, \dots, r - 1$.

DEFINITION 1.1. *Let G be a connected graph and $r \in \mathbb{N}_{>0}$. We will call a first-passage walk (FPW) of length r from i to w any walk of length r of the form*

$$i = i_0, i_1, i_2, \dots, i_{r-1}, i_r = w$$

such that $i_k \neq w$ for all $k = 0, \dots, r - 1$.

We remark that the only admissible closed FPWs are those of length $r = 0$.

The following is a standard result in graph theory and can be easily proved, e.g., by induction.

LEMMA 1.2. *The (i, j) th entry of A^r , for $r \in \mathbb{N}$, is the number of walks of length r from i to j .*

An undirected graph is said to be *connected* if, for any two nodes in G , there exists a walk of finite length connecting them, *disconnected* otherwise. Throughout this paper, we will assume that all graphs are connected, unless otherwise stated.

Remark 1.3. This simplifying assumption does not weaken the results presented in later sections. Indeed, if a graph was disconnected, then its adjacency matrix would be permutation similar to a block diagonal matrix. Our results would then apply straightforwardly to the individual diagonal blocks.

The last definition we need in this section is that of (*geodesic*) *distance* between two nodes $i, j \in V$ in a connected graph, which is defined as

$$\text{dist}(i, j) = \underset{r \in \mathbb{N}}{\text{argmin}}\{(A^r)_{ij} > 0\}.$$

Throughout this paper we will adopt the following notation; $\mathbf{1} \in \mathbb{R}^n$ will represent the vector of all ones, $\mathbf{0} \in \mathbb{R}^n$ will represent the zero vector, $I \in \mathbb{R}^{n \times n}$ will be the identity matrix, and \mathbf{e}_k will be its k th column. Further, we will denote by $\mathbf{a}_i = A\mathbf{e}_i$ the i th column of A .

1.1.1. Walk-based centrality measures. Centrality measures are a standard tool in network analysis. Customarily, they assign a nonnegative score to entities in a graph (e.g., nodes, edges) to quantify their importance. The larger the score, the more important the entity. In this paper, we are concerned with node centrality measures. Several concepts of centralities have been introduced over the years; see, e.g., [18, Chapter 7] and [34, Chapter 7]. One important class of measures is defined in terms of entries (or sums thereof) of the formal series

$$\sum_{r=0}^{\infty} b_r A^r = b_0 I + b_1 A + b_2 A^2 + \dots$$

where $b_r \geq 0$ are weights. The above expression has a nice interpretation in terms of walks taking place around the network, thanks to Lemma 1.2, and formalizes a class of measures commonly referred to as *walk-based centrality measures*; see [5, Section 3] and references therein. Depending on the choice of weights b_r and on the spectral properties of A , the formal series may converge to a matrix function $f(A)$; see, e.g., [24, Section 1.2]. In network science, the most popular choices for the weights are $b_r = \alpha^r$, for some $\alpha > 0$, and $b_r = \beta^r / (r!)$, for some $\beta > 0$, which give rise to the matrix resolvent and exponential, respectively.

In this paper, we will focus on $b_r = \alpha^r$, for some $\alpha > 0$. The induced centrality measure is known as *Katz centrality*. Introduced in [29] it assigns to node i the i th entry of the vector

$$\mathbf{x} := (I + \alpha A + \alpha^2 A^2 + \dots) \mathbf{1} = \sum_{r=0}^{\infty} \alpha^r A^r \mathbf{1}$$

where $\alpha > 0$ is a damping parameter. Selecting α so that $0 < \alpha \rho(A) < 1$, where $\rho(A)$ is the spectral radius of the adjacency matrix A , as we will do in the remainder of the

paper, ensures convergence of the above series, and thus the vector of Katz centralities $\mathbf{x} = (x_i)$ can be seen to be the solution to the (usually sparse) linear system

$$(I - \alpha A)\mathbf{x} = \mathbf{1}.$$

1.2. Problem setting and outline. The goal of this paper is to develop efficient algorithms based on the combinatorics of walks for updating the vector of Katz centrality after node/edge removal(s). We will assume that the removal of an edge $\{i, j\} \in E$ from a network corresponds to setting to zero the relevant entries of the original adjacency matrix. Similarly, the removal of node i is modelled as the removal of all the edges adjacent to it, so that the dimension of the adjacency matrix of the new graph is the same as that of the original graph. Both removal operations correspond to rank-2 modifications of A ;

$$A - [\mathbf{e}_i, \mathbf{e}_j][\mathbf{e}_j, \mathbf{e}_i]^T \quad \text{and} \quad A - [\mathbf{e}_i, \mathbf{a}_i][\mathbf{a}_i, \mathbf{e}_i]^T.$$

Given a set $\mathcal{S} \neq \emptyset$ of vertices or edges, i.e., $\mathcal{S} \subset V$ or $\mathcal{S} \subset E$, we will denote by $\mathbf{c}_r^{\mathcal{S}} \in \mathbb{R}^n$ the vector whose i th entry $(\mathbf{c}_r^{\mathcal{S}})_i$ is the number of walks of length $r > 0^1$ originating from node i , ending anywhere in the network, and visiting *at least one* of the elements in \mathcal{S} , for all $i = 1, \dots, n$. Entrywise, it holds that $A^r \mathbf{1} \geq \mathbf{c}_r^{\mathcal{S}}$. The following result follows trivially from [Lemma 1.2](#).

LEMMA 1.4. *Let $A_{\mathcal{S}}$ be the adjacency matrix of the graph obtained from G by removing all the elements in \mathcal{S} , then*

$$(1.1) \quad \mathbf{c}_r^{\mathcal{S}} = (A^r - A_{\mathcal{S}}^r)\mathbf{1}.$$

Our algorithms will build on alternative expressions for the computation of $\mathbf{c}_r^{\mathcal{S}}$, obtained using the concepts of first-passage walks (cf., [Definition 1.1](#)) and of the newly introduced \mathcal{F} -avoiding first-passage walks; see [Definition 3.3](#) below.

Remark 1.5. The formulas derived in this paper are easily adapted to the case where the entries of $\mathbf{c}_r^{\mathcal{S}} \in \mathbb{R}^n$ are set to represent walks that end at a specific (subset of) node(s) rather than at any node. To accommodate this change, one would need to replace every instance of $\mathbf{1}$ in the formulas with an indicator (one-hot encoding) vector $\mathbf{v} \in \{0, 1\}^n$ identifying the desired targets.

Remark 1.6. Since (entrywise) $0 \leq A_{\mathcal{S}} \leq A$, [[26](#), Theorem 8.1.18] guarantees that $\rho(A_{\mathcal{S}}) \leq \rho(A)$ and thus any choice of α that ensures $\sum_r \alpha^r A^r = (I - \alpha A)^{-1}$ will also imply that $\sum_r \alpha^r (A_{\mathcal{S}})^r = (I - \alpha A_{\mathcal{S}})^{-1}$.

The paper is organized as follows. The next two sections will be devoted to finding equivalent expressions for $\mathbf{c}_r^{\mathcal{S}}$ in terms of first-passage walks. These alternative formulas will allow us to describe in closed form, and in terms of “loss of walks”, the change in Katz centrality scores and related total network communicability when nodes or edges (or both) are removed from a graph; see [sections 4](#) and [5](#).

In [section 6](#) we describe two algorithms that update Katz centrality after node or edge removal in using $O(m)$ operations, where m is the number of undirected edges in the original graph. Numerical experiments are described in [section 7](#). Conclusive remarks and future work are discussed in [section 8](#)

¹When $r = 0$ we will set $\mathbf{c}_0^{\mathcal{S}} = \mathbf{0}$.

2. Counting walks through a set to edges \mathcal{E} . We begin by addressing the problem of counting walks that are forced to pass through a specific edge ($\mathcal{S} = \mathcal{E} = \{e\} \subset E$). This result will then be generalized to the case where walks have to visit at least one element in a larger set of edges, and hence, in the next section, to the problem of counting walks that visit at least one node in a set. The key simple observation that we will exploit is that all walks that visit a given edge e can be uniquely split into three parts; an initial walk that does not visit e , a step visiting edge e (in either direction), and finally another walk that roams freely around the graph.

PROPOSITION 2.1. *Let $e = \{u, v\} \in E$, $\mathcal{E} = \{e\}$, and $r > 0$. It holds that*

$$(2.1) \quad \mathbf{c}_r^{\{e\}} = \left(\sum_{k=0}^{r-1} (A_{\mathcal{E}})^k (\mathbf{e}_u \mathbf{e}_v^T + \mathbf{e}_v \mathbf{e}_u^T) A^{r-k-1} \right) \mathbf{1},$$

where $A_{\mathcal{E}} = A - \mathbf{e}_u \mathbf{e}_v^T - \mathbf{e}_v \mathbf{e}_u^T$ is the adjacency matrix of the graph obtained after the removal of e from G .

Before proving the result, we want to remark two things. Firstly, for any given $i \in V$, $\mathbf{e}_i^T (A_{\mathcal{E}})^k \mathbf{e}_u = 0$ (resp., $\mathbf{e}_i^T (A_{\mathcal{E}})^k \mathbf{e}_v = 0$) for all $k < \ell_u := \text{dist}(i, u)$ (resp., $k < \ell_v := \text{dist}(i, v)$), so that the above can be rewritten entrywise as

$$(2.2) \quad (\mathbf{c}_r^{\{e\}})_i = \mathbf{e}_i^T \left(\sum_{k=\ell_u}^{r-1} (A_{\mathcal{E}})^k \mathbf{e}_u \mathbf{e}_v^T A^{r-k-1} + \sum_{k=\ell_v}^{r-1} (A_{\mathcal{E}})^k \mathbf{e}_v \mathbf{e}_u^T A^{r-k-1} \right) \mathbf{1}.$$

Secondly, if $r \leq \text{dist}(i, e) := \min\{\ell_u, \ell_v\}$, then $r - 1 < \text{dist}(i, e)$ and thus both summations in the right-hand side of (2.2) are zero. Therefore we can assume that $r \geq \text{dist}(i, e) + 1$ whenever we are working on a specific $i \in V$.

Proof of Proposition 2.1. We proceed entrywise. Let $i \in V$ and let $\ell_x = \text{dist}(i, x)$ for $x = u, v$. Suppose wlog that $\ell_u \leq \ell_v$; if $r \leq \ell_u$, then there are no walks of length r originating from i and visiting edge e , therefore $(\mathbf{c}_r^{\mathcal{E}})_i = 0$. Similarly, for all $k = 0, \dots, r - 1$ it holds that $\mathbf{e}_i^T (A_{\mathcal{E}})^k \mathbf{e}_x = 0$ for $x = u, v$ and thus equality holds in (2.1). Suppose now that $r > \ell_u$ is fixed. Every walk of such length can be uniquely split into three consecutive terms; a walk of length $0 \leq k < r$ that does not visit e , followed by e (in either direction), and then by a walk of length $r - (k + 1)$. For any $r > \ell_u$, these can be counted using $\mathbf{e}_i^T (A_{\mathcal{E}})^k \mathbf{e}_x \mathbf{e}_y^T (A^{r-k-1} \mathbf{1})$, where either $x = u$ and $y = v$ or $x = v$ and $y = u$, depending on which between u and v is visited first. Letting k vary and splitting the terms depending on which between u or v is first visited yields the result. \square

Our formula, which allows us to count all walks originating from a node and ending at any node after visiting a given edge at least once, can be easily generalized to the situation where we are visiting at least one element in a larger set of edges.

PROPOSITION 2.2. *Let $\mathcal{E} \subset E$ be a set of edges $|\mathcal{E}| \geq 1$ and assume that, for every $e \in \mathcal{E}$, $e = \{u_e, v_e\}$. Then,*

$$\mathbf{c}_r^{\mathcal{E}} = \sum_{e \in \mathcal{E}} \mathbf{c}_r^{\{e\}} = \sum_{e \in \mathcal{E}} \left(\sum_{k=0}^{r-1} (A_{\mathcal{E}})^k (\mathbf{e}_{u_e} \mathbf{e}_{v_e}^T + \mathbf{e}_{v_e} \mathbf{e}_{u_e}^T) A^{r-k-1} \right) \mathbf{1},$$

where $A_{\mathcal{E}} = A - \sum_{e \in \mathcal{E}} (\mathbf{e}_{u_e} \mathbf{e}_{v_e}^T + \mathbf{e}_{v_e} \mathbf{e}_{u_e}^T)$ is the adjacency matrix of graph obtained from G after edges in \mathcal{E} are removed.

The proof of this result is in essence the same as that of [Proposition 2.1](#), and is therefore omitted.

Remark 2.3. From the definition of $A_{\mathcal{E}}$ it immediately follows that

$$(2.3) \quad \mathbf{c}_r^{\mathcal{E}} = \sum_{k=0}^{r-1} (A_{\mathcal{E}})^k A^{r-k} \mathbf{1} - \sum_{k=0}^{r-1} (A_{\mathcal{E}})^{k+1} A^{r-k-1} \mathbf{1}.$$

A natural extension of [Proposition 2.2](#) is presented in the next section, where we describe formulas to count walks that visit at least one node in a set $\mathcal{N} \subset V$.

3. Counting walks through a set of nodes \mathcal{N} . In this section we consider the problem of counting walks that visit at least one node in a set. This will in turn allow us to provide formulas to update the vector of Katz centrality when a graph is updated through node(s) removal; see [section 4](#). For our purposes, we identify a node w with the set of edges it connects to. This will allow us to adapt the results from the previous section to our current problem.

3.1. One node. We begin by considering the case of one node w . Let thus $\mathcal{N} = \{w\}$ and consider the corresponding set of edges

$$\mathcal{E} = \{\{w, v\} \in E : w \in \mathcal{N}\}.$$

Then, the vector $\mathbf{c}_r^{\{w\}}$ whose entries count the number of walks of length r that visit w at least once is defined as

$$\mathbf{c}_r^{\{w\}} := \mathbf{c}_r^{\mathcal{E}}$$

and moreover it holds

$$\begin{aligned} \mathbf{c}_r^{\{w\}} = \mathbf{c}_r^{\mathcal{E}} &= \sum_{\{w,v\} \in \mathcal{E}} \left(\sum_{k=0}^{r-1} (A_{\mathcal{E}})^k (\mathbf{e}_w \mathbf{e}_v^T + \mathbf{e}_v \mathbf{e}_w^T) A^{r-k-1} \right) \mathbf{1} \\ &= \sum_{v \in V} A_{vw} \left(\sum_{k=0}^{r-1} (A_{\mathcal{E}})^k (\mathbf{e}_w \mathbf{e}_v^T + \mathbf{e}_v \mathbf{e}_w^T) A^{r-k-1} \right) \mathbf{1} \\ &= \sum_{k=0}^{r-1} (A_{\mathcal{E}})^k (\mathbf{e}_w \mathbf{a}_w^T + \mathbf{a}_w \mathbf{e}_w^T) A^{r-k-1} \mathbf{1} \\ &= \sum_{k=0}^{r-1} ((A_{\mathcal{E}})^k \mathbf{e}_w \mathbf{e}_w^T A^{r-k} + (A_{\mathcal{E}})^k A \mathbf{e}_w \mathbf{e}_w^T A^{r-k-1}) \mathbf{1} \end{aligned}$$

where we exploited the fact that $\sum_{v \in V} A_{vw} \mathbf{e}_v = \mathbf{a}_w = A \mathbf{e}_w$. Now, by introducing the notation $A_{\mathcal{N}} := A_{\mathcal{E}}$ for $\mathcal{E} = \{\{w, v\} \in E : w \in \mathcal{N}\}$ and observing that

$$(A_{\mathcal{N}})^k \mathbf{e}_w = \begin{cases} \mathbf{e}_w & \text{if } k = 0, \\ \mathbf{0} & \text{if } k > 0, \end{cases}$$

the above rewrites as

$$(3.1) \quad \mathbf{c}_r^{\{w\}} = (\mathbf{e}_w^T A^r \mathbf{1}) \mathbf{e}_w + \sum_{k=0}^{r-1} (\mathbf{e}_w^T A^{r-k-1} \mathbf{1}) (A_{\mathcal{N}})^k A \mathbf{e}_w.$$

The terms $(A_{\mathcal{N}})^k \mathbf{A} \mathbf{e}_w$ appearing in (3.1) are of particular interest to us. For any given $k = 0, \dots, r-1$, the entries of this vector count the number of all those walks of length $k+1$ that consist of a walk of length k that does not visit w at any point, followed by one step to reach node w , if such step exists. In fact, these are all the first-passage walks (FPWs) from any node to w in G ; cf. Definition 1.1. Being able to count FPWs efficiently, especially when nodes are sequentially removed one by one, is at the heart of the algorithms proposed in section 6.

For any given walk-length $k \in \mathbb{N}$, we will denote by $(\mathbf{q}_k)_i$ the number of FPWs of length k originating from i (and ending at node w). For $i \neq w$, it holds that

$$(3.2) \quad (\mathbf{q}_k)_i = \begin{cases} 0 & \text{if } 0 \leq k < \text{dist}(i, w), \\ (A_{\mathcal{N}}^{k-1} \mathbf{A} \mathbf{e}_w)_i & \text{if } k \geq \text{dist}(i, w). \end{cases}$$

On the other hand, when $i = w$, we have

$$(\mathbf{q}_k)_w = \begin{cases} 1 & \text{if } k = 0, \\ 0 & \text{otherwise.} \end{cases}$$

Using this notation, (3.1) can be rewritten as

$$\mathbf{c}_r^{\{w\}} = (\mathbf{e}_w^T A^r \mathbf{1}) \mathbf{e}_w + \sum_{k=0}^{r-1} (\mathbf{e}_w^T A^{r-k-1} \mathbf{1}) \mathbf{q}_{k+1} = (\mathbf{e}_w^T A^r \mathbf{1}) \mathbf{e}_w + \sum_{k=1}^r (\mathbf{e}_w^T A^{r-k} \mathbf{1}) \mathbf{q}_k,$$

and thus we have proved the following.

PROPOSITION 3.1. *Let $\mathbf{c}_r^{\{w\}}$ be the vector whose entries count the number of walks of length r ending at any node after visiting $\mathcal{N} = \{w\}$ at least once; then*

$$(3.3) \quad \mathbf{c}_r^{\{w\}} = \sum_{k=0}^r (\mathbf{e}_w^T A^{r-k} \mathbf{1}) \mathbf{q}_k.$$

The expression in (3.3) can be written entrywise for all $i \in V$ as

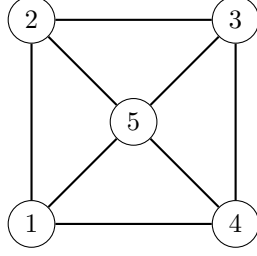
$$(\mathbf{c}_r^{\{w\}})_i = (\mathbf{e}_w^T A^r \mathbf{1}) \delta_{iw} + \sum_{k=\ell_i}^r (\mathbf{e}_w^T A^{r-k} \mathbf{1}) (\mathbf{q}_k)_i,$$

where $\ell_i = \text{dist}(i, w)$.

3.2. More than one node. We now want to consider the problem of counting walks that visit at least one node in a larger set of nodes $\mathcal{N} \subset V$ with $|\mathcal{N}| > 1$. We begin by observing that, differently from what was done when we removed multiple edges, in general we cannot simply set

$$\mathbf{c}_r^{\mathcal{N}} := \sum_{w \in \mathcal{N}} \mathbf{c}_r^{\{w\}}.$$

Indeed, consider the graph in Figure 1 and suppose that we want our walks to visit at least one node in $\mathcal{N} = \{1, 2\}$. Using the above formula, we get, for example, $(\mathbf{c}_1^{\mathcal{N}})_1 = 4$; however, the number of walks of length one originating from node 1 and visiting at least one of the nodes in \mathcal{N} is three ($1 \rightarrow 2$, $1 \rightarrow 4$, $1 \rightarrow 5$). This mismatch arises because the nodes that we want our walks to visit are connected to each other,


 FIGURE 1. *Toy network.*

causing walk $1 \rightarrow 2$ to be counted twice in the right hand side of the above expression, once by the contributing term $\mathbf{c}_1^{\{1\}}$ and once by the contributing term $\mathbf{c}_1^{\{2\}}$. This issue carries through to longer walks. Thus, a different approach should be devised that correctly counts walks when neighbouring nodes are removed. In the following, we provide two.

The first approach uses the relationships derived in the previous section to define $\mathbf{c}_r^{\mathcal{N}} := \mathbf{c}_r^{\mathcal{E}}$, where, given a set of nodes $\mathcal{N} = \{w_1, w_2, \dots, w_s\}$ with $s \geq 1$, the corresponding set of edges is

$$\mathcal{E} = \{\{w_i, j\} \in E : \text{for all } w_i \in \mathcal{N}\}.$$

Before deriving a formula for this case, we introduce some notation;

- $\mathcal{E}^{\text{in}} = \{\{v, w\} \in \mathcal{E} : v, w \in \mathcal{N}\}$ is the subset of \mathcal{E} containing edges connecting two nodes in \mathcal{N} ;
- $\mathcal{E}^{\text{out}} = \{\{w, j\} \in \mathcal{E} : w \in \mathcal{N} \text{ and } j \in V \setminus \mathcal{N}\}$ is the subset of \mathcal{E} containing edges connecting a node in \mathcal{N} with a node in $V \setminus \mathcal{N}$, i.e., it is $\mathcal{E}^{\text{out}} = \mathcal{E} \setminus \mathcal{E}^{\text{in}}$;
- A^{in} , and A^{out} are the adjacency matrices corresponding to the graphs $G^{\text{in}} = (V, \mathcal{E}^{\text{in}})$ and $G^{\text{out}} = (V, \mathcal{E}^{\text{out}})$, respectively;
- $A_{\mathcal{N}}$ denotes the adjacency matrix of the graph obtained after removing the edges in \mathcal{E} (equivalently, nodes in \mathcal{N}).

Considering again the graph in [Figure 1](#) we have that, for $\mathcal{N} = \{1, 2\}$, the sets of edges are $\mathcal{E} = \{\{1, 2\}, \{1, 4\}, \{1, 5\}, \{2, 3\}, \{2, 5\}\}$, $\mathcal{E}^{\text{in}} = \{\{1, 2\}\}$, and $\mathcal{E}^{\text{out}} = \{\{1, 4\}, \{1, 5\}, \{2, 3\}, \{2, 5\}\}$. The graphs underlying matrices $A_{\mathcal{N}}$, A^{in} , A^{out} are represented in [Figure 2](#). We point out that $A = A_{\mathcal{N}} + A^{\text{in}} + A^{\text{out}}$.

We are now in a position to derive an explicit formula for $\mathbf{c}_r^{\mathcal{N}}$ for $r > 0$. By

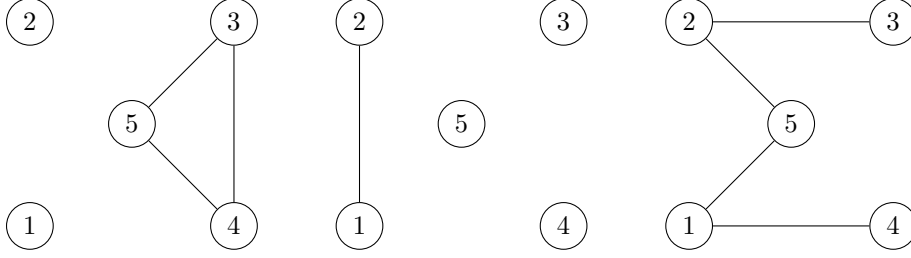


FIGURE 2. Graphs corresponding to the adjacency matrix of $A_{\mathcal{N}}$ (left), A^{in} (middle) and A^{out} (right) when $\mathcal{N} = \{1, 2\}$ from the network in Figure 1.

Proposition 2.2 it follows

$$\begin{aligned}
 \mathbf{c}_r^{\mathcal{N}} &= \sum_{e \in \mathcal{E}} \mathbf{c}_r^{\{e\}} = \sum_{e \in \mathcal{E}^{\text{in}}} \mathbf{c}_r^{\{e\}} + \sum_{e \in \mathcal{E}^{\text{out}}} \mathbf{c}_r^{\{e\}} \\
 &= \sum_{\substack{\{v,w\} \in \mathcal{E} \\ v,w \in \mathcal{N}}} \sum_{k=0}^{r-1} (A_{\mathcal{N}})^k (\mathbf{e}_v \mathbf{e}_w^T + \mathbf{e}_w \mathbf{e}_v^T) A^{r-k-1} \mathbf{1} \\
 &\quad + \sum_{\substack{\{w,j\} \in \mathcal{E} \\ w \in \mathcal{N} \\ j \in V \setminus \mathcal{N}}} \sum_{k=0}^{r-1} (A_{\mathcal{N}})^k (\mathbf{e}_w \mathbf{e}_j^T + \mathbf{e}_j \mathbf{e}_w^T) A^{r-k-1} \mathbf{1} \\
 &= \sum_{v,w \in \mathcal{N}} A_{vw} (\mathbf{e}_v \mathbf{e}_w^T + \mathbf{e}_w \mathbf{e}_v^T) A^{r-1} \mathbf{1} \\
 &\quad + \sum_{\substack{w \in \mathcal{N} \\ j \in V \setminus \mathcal{N}}} \sum_{k=0}^{r-1} (A_{\mathcal{N}})^k A_{wj} (\mathbf{e}_w \mathbf{e}_j^T + \mathbf{e}_j \mathbf{e}_w^T) A^{r-k-1} \mathbf{1}
 \end{aligned}$$

where we have used the fact that $(A_{\mathcal{N}})^k \mathbf{e}_w = \mathbf{0}$ for all $k > 0$ and for all $w \in \mathcal{N}$ and $(A_{\mathcal{N}})^k \mathbf{e}_i = \mathbf{e}_i$ for $k = 0$ and for all $i \in V$. Using now the following relationships

$$\sum_{v,w \in \mathcal{N}} A_{vw} (\mathbf{e}_v \mathbf{e}_w^T + \mathbf{e}_w \mathbf{e}_v^T) = A^{\text{in}} \quad \text{and} \quad \sum_{\substack{w \in \mathcal{N} \\ j \in V \setminus \mathcal{N}}} A_{wj} (\mathbf{e}_j \mathbf{e}_w^T + \mathbf{e}_w \mathbf{e}_j^T) = A^{\text{out}}$$

we derive

$$(3.4) \quad \mathbf{c}_r^{\mathcal{N}} = A^{\text{in}} A^{r-1} \mathbf{1} + \sum_{k=0}^{r-1} (A_{\mathcal{N}})^k A^{\text{out}} A^{r-k-1} \mathbf{1}.$$

Equation (3.4) shows that we can represent $\mathbf{c}_r^{\mathcal{N}}$ as the sum of two terms; The first counts those walks of length r that start with an edge connecting two nodes in \mathcal{E} , if such an edge exists, i.e., if \mathcal{N} contains two neighbouring nodes. The second term, on the other hand, includes walks that avoid nodes in \mathcal{N} for the first k steps, then travel in one step from a node in $V \setminus \mathcal{N}$ to a node in \mathcal{N} , and then take the remaining $r - k - 1$ steps around the network with no restrictions. We stress that this second term also includes walks where two neighbouring nodes in \mathcal{N} are consecutively visited, as long as at least one step is taken before visiting the edge connecting them.

Remark 3.2. When there are no neighbouring nodes in \mathcal{N} , i.e., when $\mathcal{E}^{\text{in}} = \emptyset$, it holds that $A^{\text{out}} = A - A_{\mathcal{N}}$ and (3.4) reduces to (2.3), i.e., $\mathbf{c}_r^{\mathcal{N}} = \sum_{w \in \mathcal{N}} \mathbf{c}_r^{\{w\}}$.

This approach shows that it is fundamental to be able to identify the first time a node in \mathcal{N} is visited by a walk, especially if two or more nodes in \mathcal{N} are connected to each other. This leads us to the introduction of the following generalization of the concept of first-passage walks, which will be at core of our second derivation of an expression for $\mathbf{c}_r^{\mathcal{N}}$; see Proposition 3.4 below.

DEFINITION 3.3. *Let $G = (V, E)$ be a graph, $i, w \in V$, $r \in \mathbb{N}$, and $\mathcal{F} \subset V \setminus \{w\}$. An \mathcal{F} -avoiding first-passage walk from i to w of length r is a FPW from i to w of length r*

$$i = i_0, i_1, i_2, \dots, i_{r-1}, i_r = w$$

such that $i_k \notin \mathcal{F}$ for all $k = 0, \dots, r-1$.

The newly introduced \mathcal{F} -avoiding FPWs will allow us to count walks that visit all nodes in \mathcal{N} while at the same time making sure that walks that visit two neighbouring nodes in \mathcal{N} successively are not counted twice. To achieve this, we will use in our count the \mathcal{F}_w -avoiding FPWs, where $\mathcal{F}_w := \mathcal{N} \setminus \{w\}$, as we let $w \in \mathcal{N}$ vary. To this end, we introduce the following notation. We define the vector $\mathbf{q}_r^{\mathcal{F}} \in \mathbb{R}^n$ as the vector whose i th entry is the number of \mathcal{F} -avoiding FPWs of length $r \in \mathbb{N}$ from i to $w \notin \mathcal{F}$. Clearly, there are no closed \mathcal{F} -avoiding FPWs unless $k = 0$, in which case $(\mathbf{q}_0^{\mathcal{F}})_i = \delta_{iw}$. Moreover, if $0 < k < \text{dist}(i, w)$ it immediately follows that $(\mathbf{q}_k^{\mathcal{F}})_i = 0$.

Thus, when $i \neq w$, it holds that

$$(3.5) \quad (\mathbf{q}_k^{\mathcal{F}})_i = \begin{cases} 0 & \text{if } 0 \leq k < \text{dist}(i, w), \\ (A_{\mathcal{N}}^{k-1} A_{\mathcal{F}} \mathbf{e}_w)_i & \text{if } k \geq \text{dist}(i, w), \end{cases}$$

where $A_{\mathcal{N}}$ and $A_{\mathcal{F}}$ are the adjacency matrices of the graphs obtained after removal of nodes in $\mathcal{N} = \mathcal{F} \cup \{w\}$ and in \mathcal{F} , respectively. We note in passing that this definition implies $(\mathbf{q}_k^{\mathcal{F}})_i = 0$ for all $i \in \mathcal{F}$ and for all $k \geq 0$. On the other hand, when $i = w$, we have

$$(\mathbf{q}_k^{\mathcal{F}})_w = \begin{cases} 1 & \text{if } k = 0, \\ 0 & \text{otherwise.} \end{cases}$$

PROPOSITION 3.4. *In the above notation, the number of walks of length $r > 0$ from $i \in V$ to any other node and passing through at least one of the nodes in \mathcal{N} is*

$$(3.6) \quad (\mathbf{c}_r^{\mathcal{N}})_i = \begin{cases} \sum_{w \in \mathcal{N}} \sum_{k=1}^r (\mathbf{q}_k^{\mathcal{F}_w})_i [A^{r-k} \mathbf{1}]_w & \text{if } i \notin \mathcal{N} \\ (A^r \mathbf{1})_i & \text{otherwise,} \end{cases}$$

where $(\mathbf{q}_k^{\mathcal{F}_w})_i$ is the number of \mathcal{F}_w -avoiding FPWs of length r from i to w , and $\mathcal{F}_w = \mathcal{N} \setminus \{w\}$.

Proof. The case $i \in \mathcal{N}$ holds trivially, as any walk originating from i is a walk that originates from $i \in \mathcal{N}$ and visits at least one node in \mathcal{N} .

Suppose now that $i \notin \mathcal{N}$, so that $\mathbf{e}_i^T A^{\text{in}} = \mathbf{0}^T$. Using the fact that

$$A^{\text{out}} = \sum_{\substack{w \in \mathcal{N} \\ j \in V \setminus \mathcal{N}}} A_{jw} (\mathbf{e}_j \mathbf{e}_w^T + \mathbf{e}_w \mathbf{e}_j^T),$$

and observing that $(A_{\mathcal{N}})^k \mathbf{e}_w = \mathbf{0}$ for all $k > 0$, (3.4) becomes

$$(\mathbf{c}_r^{\mathcal{N}})_i = \mathbf{e}_i^T \left(\sum_{k=0}^{r-1} (A_{\mathcal{N}})^k A^{\text{out}} A^{r-k-1} \mathbf{1} \right) = \sum_{k=0}^{r-1} \sum_{\substack{w \in \mathcal{N} \\ j \in V \setminus \mathcal{N}}} \mathbf{e}_i^T (A_{\mathcal{N}})^k \mathbf{e}_j A_{jw} \mathbf{e}_w^T A^{r-k-1} \mathbf{1}.$$

The walks counted in this summation start at node i , take $k \geq \text{dist}(i, j) \geq 0$ steps in nodes not in \mathcal{N} and end in a neighbor $j \in V \setminus \mathcal{N}$ of a node $w \in \mathcal{N}$. Using A^{out} , these walks take one step to reach $w \in \mathcal{N}$. They finally take $r - k - 1 \geq 0$ steps freely around the whole network. These can also be counted by considering all walks that visit one of the nodes in \mathcal{N} for the first time after $k + 1$ steps, followed by $r - k - 1$ steps around the whole network, i.e., the \mathcal{F}_w -avoiding FPW of length $k + 1$ from i to a node $w \in \mathcal{N}$ where $\mathcal{F}_w = \mathcal{N} \setminus \{w\}$, as w varies, followed by $r - k - 1$ steps around the graph. A change of variable then yields the conclusion. \square

Remark 3.5. When $\mathcal{F} = \emptyset$, Definition 3.3 becomes Definition 1.1, the relationships in (3.5) boil down to those in (3.2) with $\mathcal{N} = \{w\}$, and Proposition 3.4 reduces to Proposition 3.1.

One of the main reasons to rewrite the count in (3.4) to become (3.6) is the following result, which gives us an efficient way to compute the number of \mathcal{F} -avoiding FPWs of any given length $r > 0$ and for any given set \mathcal{F} .

PROPOSITION 3.6. *Let $\mathcal{N} = \mathcal{F} \cup \{w\}$, where $w \in V$ and $\mathcal{F} \subset V \setminus \{w\}$. Then, in the above notation,*

$$\mathbf{q}_{k+1}^{\mathcal{F}} = A \mathbf{q}_k^{\mathcal{F}} - \sum_{j \in \mathcal{N}} (\mathbf{e}_j^T A \mathbf{q}_k^{\mathcal{F}}) \mathbf{e}_j, \quad \text{for all } k = 0, 1, 2, \dots$$

Proof. Let $A_{\mathcal{N}} = A - \sum_{j \in \mathcal{N}} (\mathbf{e}_j \mathbf{a}_j^T + \mathbf{a}_j \mathbf{e}_j^T)$ be the adjacency matrix of the graph obtained from G after the removal of all the nodes in $\mathcal{N} = \mathcal{F} \cup \{w\}$, then from (3.5), for $k = 1, 2, \dots$, we get

$$\begin{aligned} \mathbf{q}_{k+1}^{\mathcal{F}} &= A_{\mathcal{N}} \mathbf{q}_k^{\mathcal{F}} \\ &= A \mathbf{q}_k^{\mathcal{F}} - \sum_{j \in \mathcal{N}} (\mathbf{e}_j \mathbf{e}_j^T A \mathbf{q}_k^{\mathcal{F}} + \mathbf{e}_j^T \mathbf{q}_k^{\mathcal{F}} \mathbf{a}_j) \\ &= A \mathbf{q}_k^{\mathcal{F}} - \sum_{j \in \mathcal{N}} (\mathbf{e}_j^T A \mathbf{q}_k^{\mathcal{F}}) \mathbf{e}_j - \sum_{j \in \mathcal{N}} (\mathbf{q}_k^{\mathcal{F}})_j \mathbf{a}_j \end{aligned}$$

The fact that $(\mathbf{q}_k^{\mathcal{F}})_j = 0$ for all $j \in \mathcal{N}$ yields the conclusion for $k = 1, 2, \dots$. When $k = 0$, a direct computation shows that

$$A \mathbf{q}_0^{\mathcal{F}} - \sum_{j \in \mathcal{F}} (\mathbf{e}_j^T \mathbf{a}_w) \mathbf{e}_j = \mathbf{a}_w - \sum_{j \in \mathcal{F}} (\mathbf{e}_j^T \mathbf{a}_w) \mathbf{e}_j,$$

which is exactly the vector of walks of length one starting at any node, ending at w , and avoiding nodes in \mathcal{F} , i.e., $\mathbf{q}_1^{\mathcal{F}}$. This concludes the proof. \square

COROLLARY 3.7. *When $\mathcal{F} = \emptyset$, for all $k = 0, 1, 2, \dots$ it holds*

$$\mathbf{q}_{k+1} = (A \mathbf{q}_k) - (\mathbf{e}_w^T A \mathbf{q}_k) \mathbf{e}_w.$$

Since $\mathbf{q}_0^{\mathcal{F}} = \mathbf{e}_w$ for any given w and any given subset $\mathcal{F} \subset V \setminus \{w\}$, Proposition 3.6 (resp., Corollary 3.7) implies that it is possible to compute $\mathbf{q}_{k+1}^{\mathcal{F}}$ (resp., \mathbf{q}_{k+1}) for $k = 0, 1, 2, \dots, r$, with r matrix vector products involving the adjacency matrix A .

In the following section we show that the results obtained so far can be used to fully characterize the variation in Katz centrality for nodes (and induced network total communicability) in networks when edges (resp., nodes) are removed from the graph. This in turn will allow us to provide bounds on the decay of total communicability and thus develop numerical strategies for their update.

4. Updating Katz centrality via walk count. We will show that the difference in Katz centrality after the removal of a set \mathcal{S} of edges or nodes can be described as a weighted sum of all walks passing through such edges or nodes, hence generalizing [10, Proposition 5.1.b] to the case in which $\mathcal{S} = \mathcal{E} \subset E$ or $\mathcal{S} = \mathcal{N} \subset V$. In the following, we will denote by $\mathbf{x}^{\mathcal{S}} = (x_i^{\mathcal{S}})$ the vector of Katz centralities computed on the graph G after removal of the elements in \mathcal{S} . We begin by proving the following simple result.

THEOREM 4.1. *In the above notation, given a set $\mathcal{N} \subset V$ of nodes and a set $\mathcal{E} \subset E$ of edges, it holds that*

$$\mathbf{x} - \mathbf{x}^{\mathcal{E}} = \sum_{r=0}^{\infty} \alpha^r \mathbf{c}_r^{\mathcal{E}} \quad \text{and} \quad \mathbf{x} - \mathbf{x}^{\mathcal{N}} = \sum_{r=0}^{\infty} \alpha^r \mathbf{c}_r^{\mathcal{N}},$$

where $\mathbf{c}_0^{\mathcal{S}} = \mathbf{0}$ and $(\mathbf{c}_r^{\mathcal{S}})_i$ is the number of walks of length $r > 0$ starting from node i and ending at any node after visiting at least one element in $\mathcal{S} = \mathcal{N}, \mathcal{E}$.

Proof. Let \mathcal{S} denote either $\mathcal{N} \subset V$ or $\mathcal{E} \subset E$, and let $A_{\mathcal{S}}$ denote the adjacency matrix of the graph obtained from G by removing all the elements in \mathcal{S} , then

$$\mathbf{x} - \mathbf{x}^{\mathcal{S}} = \sum_{r=0}^{\infty} \alpha^r A^r \mathbf{1} - \sum_{r=0}^{\infty} \alpha^r A_{\mathcal{S}}^r \mathbf{1} = \sum_{r=0}^{\infty} \alpha^r (A^r - A_{\mathcal{S}}^r) \mathbf{1}.$$

Applying (1.1) yields the result. \square

We observe that, for any given node $i \in V$, the expressions in Theorem 4.1 can be rewritten as follows

$$x_i - x_i^{\mathcal{E}} = \sum_{r=\ell_{\mathcal{E}}+1}^{\infty} \alpha^r (\mathbf{c}_r^{\mathcal{E}})_i \quad \text{and} \quad x_i - x_i^{\mathcal{N}} = \sum_{r=\ell_{\mathcal{N}}}^{\infty} \alpha^r (\mathbf{c}_r^{\mathcal{N}})_i,$$

where $\ell_{\mathcal{S}} = \min_{s \in \mathcal{S}} \{\text{dist}(i, s)\}$, with $\text{dist}(i, e) = \min\{\text{dist}(i, u), \text{dist}(i, v)\}$ for $e = \{u, v\}$. This is because there are no walks of length $k < \ell_{\mathcal{N}}$ from i to any of the elements in \mathcal{N} and there are no walks of length $k < \ell_{\mathcal{E}} + 1$ from i that visit at least one element in \mathcal{E} ; indeed, it takes $\ell_{\mathcal{E}}$ steps for a walk to reach the closest of the nodes in the closest edge $e \in \mathcal{E}$. After this, it takes one more step to traverse e .

In sections 2 and 3 we derived explicit expressions for $\mathbf{c}_r^{\mathcal{S}}$ depending on whether \mathcal{S} was a set of nodes or edges. We will use these in the following in order to provide formulas to update the vector of Katz centralities when edges or nodes are removed from a network.

4.1. Removing edges. We first consider the case in which $\mathcal{S} = \mathcal{E} \subset E$ is a set of edges.

COROLLARY 4.2. *The change in the centrality score of nodes after the removal of a set of edges \mathcal{E} is*

$$(4.1) \quad \mathbf{x} - \mathbf{x}^{\mathcal{E}} = \sum_{\{u,v\} \in \mathcal{E}} \left[\left(\sum_{k=0}^{\infty} \alpha^{k+1} (A_{\mathcal{E}})^k \mathbf{e}_v \right) x_u + \left(\sum_{k=0}^{\infty} \alpha^{k+1} (A_{\mathcal{E}})^k \mathbf{e}_u \right) x_v \right],$$

where $A_{\mathcal{E}}$ is the adjacency matrix of G after the removal of edges in \mathcal{E} . Equivalently,

$$\mathbf{x} - \mathbf{x}^{\mathcal{E}} = \alpha(I - \alpha A_{\mathcal{E}})^{-1} \sum_{\{u,v\} \in \mathcal{E}} (x_u \mathbf{e}_v + x_v \mathbf{e}_u).$$

Proof. By plugging (2.3) into Theorem 4.1, changing the order of summation, and performing a change of variables, we get

$$\begin{aligned} \mathbf{x} - \mathbf{x}^{\mathcal{E}} &= \sum_{r=0}^{\infty} \alpha^r \left(\sum_{k=0}^{r-1} (A_{\mathcal{E}})^k A^{r-k} \mathbf{1} - \sum_{k=0}^{r-1} (A_{\mathcal{E}})^{k+1} A^{r-k-1} \mathbf{1} \right) \\ &= \sum_{k=0}^{\infty} \sum_{r=k+1}^{\infty} \alpha^r (A_{\mathcal{E}})^k (A - A_{\mathcal{E}}) A^{r-k-1} \mathbf{1} \\ &= \sum_{k=0}^{\infty} \sum_{r=0}^{\infty} \alpha^{r+k+1} (A_{\mathcal{E}})^k \left(\sum_{\{u,v\} \in \mathcal{E}} (\mathbf{e}_u \mathbf{e}_v^T + \mathbf{e}_v \mathbf{e}_u^T) \right) A^r \mathbf{1}. \end{aligned}$$

Recalling that $x_i = \mathbf{e}_i^T (I - \alpha A)^{-1} \mathbf{1}$, a simple algebraic manipulation yields the conclusion. \square

COROLLARY 4.3. When $\mathcal{E} = \{e\} \subset E$ with $e = \{u, v\}$, for all $i \in V$ it holds

$$x_i - x_i^{\{e\}} = \left(\sum_{k=\ell_w}^{\infty} \alpha^{k+1} (A_{\mathcal{E}}^k)_{iv} \right) x_u + \left(\sum_{k=\ell_u}^{\infty} \alpha^{k+1} (A_{\mathcal{E}}^k)_{iu} \right) x_v,$$

where $\ell_w = \text{dist}(i, w)$ for $w = u, v$, and $A_{\mathcal{E}} = A - \mathbf{e}_u \mathbf{e}_v^T - \mathbf{e}_v \mathbf{e}_u^T$ is the adjacency matrix of graph G after the removal of edge e . Equivalently,

$$\mathbf{x} - \mathbf{x}^{\{e\}} = \alpha(I - \alpha A_{\mathcal{E}})^{-1} (x_u \mathbf{e}_v + x_v \mathbf{e}_u).$$

Corollary 4.3 allows us to provide an upper bound on the relative decrease of Katz centrality of nodes u and v when edge $e = \{u, v\}$ is removed from the graph. This is described in the following result.

PROPOSITION 4.4. Let $\mathcal{E} = \{e\} \subset E$ with $e = \{u, v\}$. Then

$$\frac{x_u^{\{e\}}}{x_u} \leq 1 - \alpha \frac{x_v}{x_u}, \quad \text{and} \quad \frac{x_v^{\{e\}}}{x_v} \leq 1 - \alpha \frac{x_u}{x_v}.$$

Moreover, if u is a leaf in G , i.e., if $\deg(u) = 1^2$, then

$$\frac{x_u^{\{e\}}}{x_u} = 1 - \alpha \frac{x_v}{x_u}.$$

Proof. We prove the first inequality. The second follows in a similar way. From Corollary 4.3, since $\ell_u = \text{dist}(u, u) = 0$ and $\ell_v = \text{dist}(u, v) = 1$, and using the fact that $A_{\mathcal{E}} \geq 0$ one gets

$$x_u - x_u^{\{e\}} = x_u \sum_{r=1}^{\infty} \alpha^{r+1} (A_{\mathcal{E}}^r)_{uv} + x_v \sum_{r=0}^{\infty} \alpha^{r+1} (A_{\mathcal{E}}^r)_{uu} \geq x_v \alpha (A_{\mathcal{E}}^0)_{uu} = \alpha x_v.$$

²Recall that G is connected and thus at most one of the two nodes can be a leaf, unless G is the path with two nodes.

The conclusion thus follows from trivial algebraic manipulations, since $x_u \geq 1$. Furthermore, if u is a leaf connected to node v via the edge that is being removed, then $(A_{\mathcal{E}}^r)_{uv} = (A_{\mathcal{E}}^r)_{uu} = 0$ for all $r \geq 1$. When $r = 0$, it holds $(A_{\mathcal{E}}^0)_{uu} = 1$ and thus the conclusion. \square

4.2. Removing nodes. We consider now the situation where a set of nodes \mathcal{N} is removed from the graph. Similarly to [section 3](#), the removal of one or more vertices will be modelled as the removal of the edges incident to them, resulting in the nodes in \mathcal{N} becoming isolated nodes in the new graph.

COROLLARY 4.5. *Let $\mathcal{N} \subset V$ be a set of nodes in G , then*

$$(4.2) \quad x_i - x_i^{\mathcal{N}} = \begin{cases} \sum_{w \in \mathcal{N}} (x_w \sum_{r=\ell_w}^{\infty} \alpha^r (\mathbf{q}_r^{\mathcal{F}_w})_i), & \text{if } i \notin \mathcal{N} \\ x_i - 1 & \text{otherwise} \end{cases}$$

where $(\mathbf{q}_r^{\mathcal{F}_w})_i$ is the number of \mathcal{F}_w -avoiding FPWs from i to w , with $\mathcal{F}_w := \mathcal{N} \setminus \{w\}$, and $\ell_w = \text{dist}(i, w)$.

Proof. The result follows, with some algebraic manipulation, from [Theorem 4.1](#) and [Proposition 3.4](#), using the fact that $(\mathbf{q}_r^{\mathcal{F}_w})_i = 0$ for all $r < \ell_w = \text{dist}(i, w)$ and the convention, set at the beginning of the paper, that $\mathbf{c}_0^{\mathcal{N}} = \mathbf{0}$. \square

COROLLARY 4.6. *In the hypothesis of [Corollary 4.5](#), if $\mathcal{N} = \{w\}$ then*

$$(4.3) \quad x_i - x_i^{\{w\}} = \begin{cases} x_w \sum_{r=\ell}^{\infty} \alpha^r (\mathbf{q}_r)_i, & \text{if } i \neq w \\ x_i - 1 & \text{otherwise,} \end{cases}$$

where $\ell = \text{dist}(i, w)$.

[Corollary 4.6](#) provides an estimate on the decrease rate of Katz centrality of node $i \sim w$ after the removal of w from the graph. Indeed, the following result holds.

PROPOSITION 4.7. *If $i \sim w$ then*

$$\frac{x_i^{\{w\}}}{x_i} \leq 1 - \alpha \frac{x_w}{x_i}.$$

Proof. Using [\(4.3\)](#) and the fact that $\text{dist}(i, w) = 1$ and $(\mathbf{q}_1)_i = 1$ since the graph is simple, one gets

$$x_i - x_i^{\{w\}} = x_w \sum_{r=1}^{\infty} \alpha^r (\mathbf{q}_r)_i \geq \alpha x_w.$$

Dividing both sides by $x_i > 1$ and rearranging yields the conclusion. \square

5. Bounds on total network communicability. In the previous section we described closed formulas that allow to compute the change in Katz centrality of nodes when edges or nodes are removed from a given simple connected graph.

In this section we use the results from [Corollary 4.3](#) and [Corollary 4.6](#) to find an upper bound on the change of the *total network communicability* [6], here defined as

$$TC = \frac{\mathbf{1}^T \mathbf{x}}{n},$$

after the removal of a node or an edge from a graph. Throughout this section, we will denote by $TC^{\mathcal{S}}$ the total communicability of the graph obtained from G after removal of elements in set \mathcal{S} .

THEOREM 5.1. Given $\mathcal{N} = \{w\} \subset V$ it holds that

$$(5.1) \quad TC - TC^{\{w\}} \leq \frac{1}{n}(x_w^2(1 - \alpha^2 \deg(w)) - 1),$$

where $\deg(w)$ is the degree of node w .

Proof. Let us consider the set of edges $\mathcal{E} = \{\{w, v\} \in E : w \in \mathcal{N}\}$. From [Corollary 4.2](#) and symmetry of $A_{\mathcal{E}}$, it follows that

$$\mathbf{1}^T(\mathbf{x} - \mathbf{x}^{\mathcal{E}}) = \alpha \mathbf{1}^T(I - \alpha A_{\mathcal{E}})^{-1} \sum_{v: v \sim w} (x_w \mathbf{e}_v + x_v \mathbf{e}_w) = \alpha x_w \sum_{v \sim w} x_v^{\mathcal{E}} + \alpha x_w^{\mathcal{E}} \sum_{v \sim w} x_v.$$

Since, by definition of Katz centrality, $x_w^{\mathcal{E}} = 1$ and

$$(5.2) \quad x_w = 1 + \alpha \sum_{v \sim w} x_v,$$

we get

$$\mathbf{1}^T(\mathbf{x} - \mathbf{x}^{\mathcal{E}}) = \alpha x_w \sum_{v \sim w} x_v^{\mathcal{E}} + (x_w - 1)$$

The fact that $\mathbf{x}^{\mathcal{E}} = \mathbf{x}^{\{w\}}$, [Proposition 4.7](#), and (5.2) imply

$$\begin{aligned} \mathbf{1}^T(\mathbf{x} - \mathbf{x}^{\{w\}}) &= x_w \left(1 + \alpha \sum_{v \sim w} x_v \left(\frac{x_v^{\{w\}}}{x_v} \right) \right) - 1 \\ &\leq x_w \left(1 + \alpha \sum_{v \sim w} x_v \left(1 - \alpha \frac{x_w}{x_v} \right) \right) - 1 \\ &= x_w (x_w - \alpha^2 x_w \deg(w)) - 1 \\ &= x_w^2 (1 - \alpha^2 \deg(w)) - 1, \end{aligned}$$

from which (5.1) immediately follows. \square

THEOREM 5.2. Let $e = \{u, v\}$ and consider $\mathcal{E} = \{e\}$. Then

$$(5.3) \quad TC - TC^{\{e\}} \leq \frac{1}{n} (2\alpha x_u x_v - \alpha^2 x_u^2 - \alpha^2 x_v^2).$$

Proof. From [Corollary 4.3](#) and symmetry of $A_{\mathcal{E}}$ we obtain

$$\mathbf{1}^T(\mathbf{x} - \mathbf{x}^{\{e\}}) = \alpha \mathbf{1}^T(I - \alpha A_{\mathcal{E}})^{-1} (x_u \mathbf{e}_v + x_v \mathbf{e}_u) = \alpha x_u x_v^{\{e\}} + \alpha x_v x_u^{\{e\}}.$$

Now by applying [Proposition 4.4](#) we get

$$\begin{aligned} \mathbf{1}^T(\mathbf{x} - \mathbf{x}^{\{e\}}) &= \alpha x_u x_v \left(\frac{x_v^{\{e\}}}{x_v} + \frac{x_u^{\{e\}}}{x_u} \right) \\ &\leq \alpha x_u x_v \left(1 - \alpha \frac{x_u}{x_v} + 1 - \alpha \frac{x_v}{x_u} \right) \\ &= 2\alpha x_u x_v - \alpha^2 x_u^2 - \alpha^2 x_v^2, \end{aligned}$$

from which the conclusion straightforwardly follows. \square

Remark 5.3. Consider now the function $f_\alpha(x, y) = 2\alpha xy - \alpha^2 x^2 - \alpha^2 y^2$ appearing in (5.3) and assume that $0 < \alpha < 1$. Let $T > t > 1$, then it can be showed that

$$\operatorname{argmin}_{(x,y) \in [t,T] \times [t,T]} f_\alpha(x, y) = \begin{cases} (t, t) & \text{if } T < \frac{2-\alpha}{\alpha}t \\ (t, T) & \text{otherwise.} \end{cases}$$

Hence, the bounds in [Theorem 5.2](#) can be used to devise strategies for optimizing total communicability through edge removal, similar to the ones in [2]. Notably, if $T < \frac{2-\alpha}{\alpha}t$, we find that

$$\operatorname{argmin}_{(x,y) \in [t,T] \times [t,T]} f_\alpha(x, y) = \operatorname{argmin}_{(x,y) \in [t,T] \times [t,T]} xy = (t, t),$$

which suggests that, in order to minimize the impact on total network communicability of an edge removal in a network such that $\max_i x_i < \frac{2-\alpha}{\alpha} \min_i x_i$, one should remove the edge $e = \{u, v\}$ that minimizes the product $x_u x_v$. This was in fact one of the heuristics proposed in [2] for solving the *downdate problem* considered there.

6. Walk-induced update of Katz centrality. In this section, we propose algorithms to find an approximate value of Katz centrality for the nodes in a simple graph G after elements from a chosen set \mathcal{S} are removed. The results presented here build on the theory developed so far, and assume that the values of the original Katz centralities, stored in a vector \mathbf{x} , are available. The algorithms developed in this section are especially useful in the context where the elements of \mathcal{S} are *sequentially* removed from G . This would normally require solving $|\mathcal{S}|$ sparse linear systems, which can be burdensome when n is large. On the other hand, the methods we propose will allow updating the centrality vectors with *approximations* to the actual values, and only sporadically solving a sparse linear system. As we will show, this reduces the computational effort of finding Katz centrality when nodes or edges are sequentially removed from G , while at the same time achieving good approximations of the new centrality scores.

Throughout this section, approximate values will be denoted by $\hat{\mathbf{x}}$.

6.1. Removing nodes. We begin with the problem of updating the vector of Katz centralities after a node is removed from G . The strategy we propose builds on the result presented in [Corollary 4.6](#) and simply truncates the infinite sum to L terms, where L is a user selected parameter. The proposed approximation, for $\mathcal{N} = \{w\}$, thus is

$$\hat{\mathbf{x}}^{\{w\}} := \mathbf{x} - x_w \sum_{r=1}^L \alpha^r \mathbf{q}_r,$$

where $(\mathbf{q}_r)_i$ is the number of FPWs of length r from node i to w . Moreover, since Katz centrality of isolated nodes is equal to one by definition, we will also have that $\hat{x}_w^{\{w\}} = x_w^{\{w\}} = 1$.

A pseudocode for our proposed algorithm can be found in [Algorithm 6.1](#).

Given that \mathbf{x} , the vector of Katz centralities, is known, the computational effort to generate the approximant $\hat{\mathbf{x}}^{\{w\}}$ is spent computing the vectors \mathbf{q}_r for $r = 1, \dots, L$. Thanks to the result presented in [Corollary 3.7](#), this operation only requires L matrix-vector products involving the adjacency matrix A . Therefore, the choice of L is crucial to obtain both a good approximation and a reasonable computational cost. The latter, in particular, should be smaller than the cost of computing Katz centrality from scratch. In our experiments we select L *dynamically*. After selecting the maximum

Algorithm 6.1 Updating Katz centrality after node removal

Input: $A \in \mathbb{R}^{n \times n}$ adjacency matrix, $\mathbf{x} \in \mathbb{R}^n$ vector of Katz centrality, $\alpha \in (0, 1/\rho(A))$, $L_{\max}^{\mathcal{N}} \in \mathbb{N}$, $\text{tol} \in \mathbb{R}$, w node to be removed.

Output: $\widehat{\mathbf{x}}^{\{w\}} \in \mathbb{R}^n$ approximate update, $L \in \mathbb{N}$ number of iterations to converge.

- 1: $\widehat{\mathbf{x}}^{\{w\}} = \mathbf{x}$;
 - 2: $L = 1$;
 - 3: $\mathbf{q} = \alpha * A * \mathbf{e}_w$;
 - 4: $\widehat{\mathbf{x}}^{\{w\}} = \widehat{\mathbf{x}}^{\{w\}} - x_w * \mathbf{q}$;
 - 5: **while** $x_w \|\mathbf{q}\|_2 / \|\mathbf{x}\|_2 > \text{tol}$ **and** $L < L_{\max}^{\mathcal{N}}$ **do**
 - 6: $\mathbf{q} = \alpha * A * \mathbf{q}$;
 - 7: $q_w = 0$;
 - 8: $\widehat{\mathbf{x}}^{\{w\}} = \widehat{\mathbf{x}}^{\{w\}} - x_w * \mathbf{q}$;
 - 9: $L = L + 1$;
 - 10: **end while**
 - 11: $(\widehat{\mathbf{x}}^{\{w\}})_w = 1$
-

Algorithm 6.2 Updating Katz centrality after edge removal

Input: $A \in \mathbb{R}^{n \times n}$ adjacency matrix, $\mathbf{x} \in \mathbb{R}^n$ vector of Katz centrality, $\alpha \in (0, 1/\rho(A))$, $L_{\max}^{\mathcal{E}} \in \mathbb{N}$, $\text{tol} \in \mathbb{R}$, $e = \{u, v\}$ edge to be removed.

Output: $\widehat{\mathbf{x}}^{\{e\}} \in \mathbb{R}^n$ approximate update, $L \in \mathbb{N}$ number of iterations to convergence.

- 1: $\widehat{\mathbf{x}}^{\{e\}} = \mathbf{x} - \alpha * x_v * \mathbf{e}_u - \alpha * x_u * \mathbf{e}_v$;
 - 2: $L = 1$;
 - 3: $\mathbf{s} = \alpha^2 * (A * \mathbf{e}_u - \mathbf{e}_v)$;
 - 4: $\mathbf{t} = \alpha^2 * (A * \mathbf{e}_v - \mathbf{e}_u)$;
 - 5: $\widehat{\mathbf{x}}^{\{e\}} = \widehat{\mathbf{x}}^{\{e\}} - x_v * \mathbf{s} - x_u * \mathbf{t}$;
 - 6: **while** $\|x_v * \mathbf{s} + x_u * \mathbf{t}\| / \|\mathbf{x}\|_2 > \text{tol}$ **and** $L < L_{\max}^{\mathcal{E}}$ **do**
 - 7: $\mathbf{s} = \alpha * (A * \mathbf{s} - s_u * \mathbf{e}_v - s_v * \mathbf{e}_u)$;
 - 8: $\mathbf{t} = \alpha * (A * \mathbf{t} - t_u * \mathbf{e}_v - t_v * \mathbf{e}_u)$;
 - 9: $\widehat{\mathbf{x}}^{\{e\}} = \widehat{\mathbf{x}}^{\{e\}} - x_v * \mathbf{s} - x_u * \mathbf{t}$;
 - 10: $L = L + 1$;
 - 11: **end while**
-

walk-length that we want to allow, denoted by $L_{\max}^{\mathcal{N}}$, the parameter L is chosen as the smallest $0 < L \leq L_{\max}^{\mathcal{N}}$ such that

$$\alpha^L x_w \frac{\|\mathbf{q}_L\|_2}{\|\mathbf{x}\|_2} < \text{tol}.$$

This is equivalent to checking that the (relative) norm of the difference between two subsequent approximations, obtained by truncating the series at $L - 1$ and L , respectively, is smaller than a selected tolerance tol .

6.2. Removing edges. For the case where an edge is removed, similarly to what we have just done for the case of nodes, we exploit the result in [Corollary 4.3](#) to derive an approximation to $\mathbf{x}^{\{e\}}$;

$$\widehat{\mathbf{x}}^{\{e\}} := \mathbf{x} - x_u \sum_{r=0}^L \alpha^{r+1} (A_{\mathcal{E}})^r \mathbf{e}_v - x_v \sum_{r=0}^L \alpha^{r+1} (A_{\mathcal{E}})^r \mathbf{e}_u,$$

where $A_{\mathcal{E}} = A - \mathbf{e}_u \mathbf{e}_v^T - \mathbf{e}_v \mathbf{e}_u^T$ is the adjacency matrix after the removal of edge $e = \{u, v\}$.

As in the previous case, we set a maximum walk length $L_{\max}^{\mathcal{E}}$ and then select $0 < L \leq L_{\max}^{\mathcal{E}}$ dynamically as the smallest index such that

$$\alpha^{L+1} \left(x_u \frac{\|A_{\mathcal{E}}^L \mathbf{e}_v\|_2}{\|\mathbf{x}\|_2} + x_v \frac{\|A_{\mathcal{E}}^L \mathbf{e}_u\|_2}{\|\mathbf{x}\|_2} \right) < \text{tol}.$$

Algorithm 6.2 contains the pseudocode for the proposed procedure.

6.3. Computational cost. The following holds.

PROPOSITION 6.1. *For a sparse network, the cost of Algorithms 6.1 and 6.2 is $\mathcal{O}(Lm)$ operations, where m is the number of undirected edges in the original graph.*

Proof. For **Algorithm 6.1**, at every step of the loop we perform a matrix-vector product on line 6 and n multiplications on lines 6 and 8. Thus, the computational complexity of **Algorithm 6.1** can be estimated as $\mathcal{O}(LC + 2Ln)$ operations, where C is the cost of performing one matrix-vector product. Similarly, one can deduce that **Algorithm 6.2** requires $\mathcal{O}(2LC + 4Ln)$ operations. The conclusion follows by recalling the reasonable assumption that $C = \mathcal{O}(m)$ for sparse matrices with $2m$ nonzero entries. \square

Clearly, **Algorithms 6.1** and **6.2** can be generalized to the case where a larger set of edges (resp., nodes) is *simultaneously* removed from the graph. To do so, it is sufficient to truncate the infinite series in (4.1) (resp., (4.2)) to L terms. In this case it is easy to see that, under the same assumption of **Proposition 6.1**, the generalized algorithms would have a computational complexity in time of $\mathcal{O}(|\mathcal{S}|Lm)$, where \mathcal{S} is the set of elements being removed. In addition, since (4.1) and (4.2) are both described as sums of $|\mathcal{S}|$ independent terms, one could use a parallel implementation on, e.g., $|\mathcal{S}|$ processors, to compute each term independently, hence achieving the theoretical computational complexity of $\mathcal{O}(Lm)$ per processor.

7. Numerical experiments. In this section, we compare the performance of **Algorithms 6.1** and **6.2** with other methods that, in contrast, simply recompute the centrality scores from scratch. In our implementation the damping factor in Katz centrality was set to $\alpha = 0.85/\rho(A)$, where $\rho(A)$ is the spectral radius of A [6]. In this section, we only considered connected networks for simplicity.

We performed experiments on both synthetic and real-world networks. For the synthetic ones, we generated 30 instances of networks with n nodes using the functions `pref(n,5)` and `erdrey(n,5n)` from the CONTEST Toolbox [23]. We refer the interested reader to the original reference for further details on parameter selection. Some simple statistics on the generated graphs, including number of nodes n , average number of edges m , average diameter diam , average eccentricity ϵ , and average condition number $\kappa(I - \alpha A)$ of $(I - \alpha A)$ are reported in **Table 1**. Recall that the diameter of a network is $\text{diam} = \max_{i,j} \text{dist}(i,j)$, while the average eccentricity is $\epsilon = \frac{1}{n} \sum_i \max_{j \in V} \{\text{dist}(i,j)\}$.

The real-world networks in our dataset, both available at [14], are a road network of the Minnesota state (`minnesota`) and a communication network between a group of Autonomous Systems (AS) in the period between 1997-2000 (`as-735`). A summary of these networks' basic features is provided in **Table 2**.

All experiments were run via MATLAB R2023b on a laptop PC equipped with an Intel i7-1260P CPU with a clock rate of 2.10 GHz, and 16 GB of RAM.

Updating Katz centrality by counting walks

TABLE 1

Synthetic networks data: number of nodes n , number of edges m , network diameter diam , average eccentricity ϵ , and condition number $\kappa(I - \alpha A)$ of $I - \alpha A$ for $\alpha = 0.85/\rho(A)$. All the values are averaged over 30 runs.

Network	n	m	diam	ϵ	$\kappa(I - \alpha A)$
erdrey($n, 5n$)	3200	16000	6.1	5.2	10
	6400	32000	6.9	5.7	10
	12800	64000	7.2	6.0	10
	25600	128000	8.0	6.5	10
pref($n, 5$)	3200	15800	5.0	4.7	10.1
	6400	31800	5.7	4.8	11.1
	12800	63700	6.0	4.9	11.6
	25600	127000	6.0	5.2	11.7

TABLE 2

Real-world network data: number of nodes n , number of edges m , network diameter diam , average eccentricity ϵ , and condition number $\kappa(I - \alpha A)$, for $\alpha = 0.85/\rho(A)$.

Network	n	m	diam	ϵ	$\kappa(I - \alpha A)$
minnesota	2640	3302	99	71.7	12.2
as-735	6474	13895	9	6.7	11.6

In the first experiment, we considered the synthetic networks in our dataset and removed one element s , either a node or an edge, selected uniformly at random from either V or E . After constructing the adjacency matrix A_S of the new graph, we compute/approximate the updated Katz centrality \mathbf{x}^S in three different ways;

- (i) **pcg** ($\mathbf{x}^{(0)} = \mathbf{v}$): solve $(I - \alpha A_S)\mathbf{y} = \mathbf{1}$ using MATLAB built-in function **pcg** (conjugate gradient) without preconditioning, initial guess $\mathbf{x}^{(0)} = \mathbf{v}$, and with a relative tolerance of 10^{-5} ;
- (ii) method in [20]: truncate the series $\sum_{r=0}^{\infty} (A_S)^r \mathbf{1}$ with a tolerance of 10^{-4} on the stopping criteria and a maximum number of iteration of 100; see [20] for details; and
- (iii) **Algorithm 6.1** or **6.2** with $L_{\max}^S = 30$ and $\text{tol} = 10^{-4}$.

Before commenting on the experimental results, some additional discussion on the stopping criteria for the above is in order. Methods (ii) and (iii) share the same stopping criterion: they stop as soon as the difference between the results of two subsequent iterations is smaller than a relative tolerance. Hence, in order to compare the two methods, it is sufficient to choose the same tolerance $\text{tol} = 10^{-4}$. For **pcg**, on the other hand, the stopping criterion checks the norm of the residual, and not the difference between two iterates. Nevertheless, we note that the norm of the difference between two subsequent iterations of the converging sequence of approximations $\{\mathbf{x}^{(k)}\}$ for the solution \mathbf{x}^* can be bounded by a quantity proportional to the norm of the residual of the underlying linear system. More precisely, we recall the following.

PROPOSITION 7.1. *Let $\{\mathbf{x}^{(k)}\}$ be a sequence converging monotonically to \mathbf{x}^* for $k > N$, solution of the nonsingular system $M\mathbf{x} = \mathbf{b}$. Then,*

$$\frac{\|\mathbf{x}^{(k-1)} - \mathbf{x}^{(k)}\|}{\|\mathbf{x}^*\|} \leq 2\kappa(M) \frac{\|\mathbf{r}^{(k-1)}\|}{\|\mathbf{b}\|},$$

where $\mathbf{r}^{(k-1)} = M\mathbf{x}^{(k-1)} - \mathbf{b}$ is the residual at the $(k-1)$ th iteration.

Updating Katz centrality by counting walks

TABLE 3

Number of iterations and CPU timings required for approximating \mathbf{x}^S with methods (i)-(iii) after the removal of one node selected uniformly at random. The chosen tolerances are $\text{tol}=10^{-5}$ for method (i), and $\text{tol}=10^{-4}$ for methods (ii)-(iii). The initial guess for pcg is specified in brackets. All the values are averaged over 30 runs.

Network	n	$\text{pcg}(\mathbf{x}^{(0)} = \mathbf{0})$		$\text{pcg}(\mathbf{x}^{(0)} = \mathbf{x})$		method in [20]		Algorithm 6.1	
		iter	time(s)	iter	time(s)	iter	time(s)	L	time(s)
erdrey(n, 5n)	3200	11.0	$1.18 \cdot 10^{-3}$	7.5	$1.08 \cdot 10^{-3}$	45.0	$1.75 \cdot 10^{-3}$	6.5	$1.27 \cdot 10^{-3}$
	6400	11.0	$6.68 \cdot 10^{-3}$	7.2	$4.29 \cdot 10^{-2}$	45.0	$5.63 \cdot 10^{-3}$	4.6	$1.08 \cdot 10^{-3}$
	12800	11.0	$1.75 \cdot 10^{-2}$	7.0	$1.28 \cdot 10^{-2}$	45.0	$3.39 \cdot 10^{-2}$	4.1	$2.20 \cdot 10^{-3}$
	25600	11.0	$2.88 \cdot 10^{-2}$	6.7	$1.99 \cdot 10^{-2}$	45.0	$5.36 \cdot 10^{-2}$	3.7	$2.47 \cdot 10^{-3}$
pref(n, 5)	3200	11.4	$1.11 \cdot 10^{-3}$	5.9	$9.28 \cdot 10^{-4}$	45.0	$1.74 \cdot 10^{-3}$	4.8	$8.32 \cdot 10^{-4}$
	6400	11.8	$6.64 \cdot 10^{-3}$	5.3	$3.30 \cdot 10^{-3}$	45.0	$4.78 \cdot 10^{-3}$	3.8	$1.09 \cdot 10^{-3}$
	12800	12.1	$1.75 \cdot 10^{-2}$	4.1	$8.57 \cdot 10^{-3}$	45.0	$3.21 \cdot 10^{-2}$	2.6	$1.38 \cdot 10^{-3}$
	25600	12.8	$3.10 \cdot 10^{-2}$	3.2	$1.28 \cdot 10^{-2}$	45.0	$5.18 \cdot 10^{-2}$	2.5	$2.42 \cdot 10^{-3}$

TABLE 4

Number of iterations and CPU timings required for approximating \mathbf{x}^S with methods (i)-(iii) after the removal of one edge selected uniformly at random. The chosen tolerances are $\text{tol}=10^{-5}$ for method (i), and $\text{tol}=10^{-4}$ for methods (ii)-(iii). The initial guess for pcg is specified in brackets. All the values are averaged over 30 runs.

Network	n	$\text{pcg}(\mathbf{x}^{(0)} = \mathbf{0})$		$\text{pcg}(\mathbf{x}^{(0)} = \mathbf{x})$		method in [20]		Algorithm 6.2	
		iter	time(s)	iter	time(s)	iter	time(s)	L	time(s)
erdrey(n, 5n)	3200	11.0	$1.19 \cdot 10^{-3}$	6.1	$1.06 \cdot 10^{-3}$	45.0	$1.71 \cdot 10^{-3}$	2.9	$5.36 \cdot 10^{-4}$
	6400	11.0	$5.19 \cdot 10^{-3}$	5.8	$3.35 \cdot 10^{-3}$	45.0	$5.49 \cdot 10^{-3}$	2.4	$4.56 \cdot 10^{-4}$
	12800	11.0	$1.38 \cdot 10^{-2}$	5.5	$8.65 \cdot 10^{-3}$	45.0	$2.50 \cdot 10^{-2}$	2.1	$7.75 \cdot 10^{-4}$
	25600	11.0	$2.85 \cdot 10^{-2}$	5.1	$1.68 \cdot 10^{-2}$	45.0	$5.08 \cdot 10^{-2}$	2.0	$1.14 \cdot 10^{-3}$
pref(n, 5)	3200	11.4	$1.08 \cdot 10^{-3}$	4.1	$8.69 \cdot 10^{-4}$	45.0	$1.71 \cdot 10^{-3}$	2.0	$5.44 \cdot 10^{-4}$
	6400	11.7	$7.19 \cdot 10^{-3}$	3.0	$2.78 \cdot 10^{-3}$	45.0	$4.84 \cdot 10^{-3}$	1.3	$3.01 \cdot 10^{-4}$
	12800	12.2	$1.81 \cdot 10^{-2}$	2.5	$6.44 \cdot 10^{-3}$	45.0	$3.21 \cdot 10^{-2}$	1.1	$5.57 \cdot 10^{-4}$
	25600	12.5	$3.06 \cdot 10^{-2}$	2.2	$1.09 \cdot 10^{-2}$	45.0	$4.80 \cdot 10^{-2}$	1.1	$1.16 \cdot 10^{-3}$

In view of [Proposition 7.1](#), in our experiments we used a relative tolerance of $\text{tol}_{\text{pcg}} := \text{tol}/10 = 10^{-5}$ for the pcg algorithm in method (i), where $\text{tol} = 10^{-4}$ is the tolerance set for methods (ii)-(iii) and the denominator comes from the observed $\kappa(I - \alpha A) \approx 10$ for all the network instances considered here; cf. [Table 1](#).

Lastly, we want to mention in passing that we do not compare our proposed strategy with the one described in [\[32\]](#) since experiments not shown here returned results comparable with those obtained by $\text{pcg}(\mathbf{x}^{(0)} = \mathbf{0})$.

In [Tables 3](#) and [4](#), we report the average number of iterations and average CPU times needed to compute an approximation to \mathbf{x}^S using the methods described. All values are averaged over 30 runs with the same parameters.

The results show that our strategy (iii) requires fewer iterations to converge, and thus smaller CPU times, compared to the other methods considered, especially in the case of large networks. Moreover, the number of iterations required to achieve convergence by methods (i) with $\mathbf{v} = \mathbf{0}$ and (ii) tends to increase with the size of the network; on the other hand, it decreases with n for method (iii) and for $\text{pcg}(\mathbf{x}^{(0)} = \mathbf{x})$. This behaviour can be explained by observing that, while $\text{pcg}(\mathbf{x}^{(0)} = \mathbf{0})$ and series truncation are iterative methods starting from a poor initial guess, namely $\mathbf{x}^{(0)} = \mathbf{0}$, our approach (iii) and $\text{pcg}(\mathbf{x}^{(0)} = \mathbf{x})$ are initialized with $\mathbf{x}^{(0)} = \mathbf{x}$, which may already be a good approximation to \mathbf{x}^S . In fact, thanks to [Corollaries 4.3](#) and [4.6](#), we know that if the removed node (resp., edge) has (resp., joins nodes with) a relative small

centrality score(s), then \mathbf{x}^S is almost identical to \mathbf{x} . An alternative explanation may also be given, for method (iii) only, in terms of the combinatorics of walks. Indeed, methods (i) with $\mathbf{v} = \mathbf{0}$ and (iii) compute \mathbf{x}^S by counting from scratch all the walks that *avoid* s , which are the ones not effected by the removal; method (iii), on the other hand, focuses on the walks that *visit* s . The observed phenomenon may be hence explained because the number of these latter walks relative to network size decreases with an increasing n (except for specific examples such as star graphs) much faster than the number of those that avoid s relative to n .

In our second set of experiments, we investigate the accuracy of [Algorithms 6.1](#) and [6.2](#) when a set of nodes (resp., edges) is removed from the graph. We sequentially removed $\lceil n/100 \rceil$ nodes (resp., $\lceil m/100 \rceil$ edges) from the synthetic networks in our dataset. After each removal, we compute both the actual value of Katz centrality and its approximation. We perform both a random and a targeted selection of nodes/edges, which simulate random failures and malicious attacks to the networks, respectively. The results are averaged over 30 runs.

To assess the effectiveness of the algorithms, we consider the relative 2-norm error $\|\mathbf{x} - \widehat{\mathbf{x}}\|/\|\mathbf{x}\|$ between the actual vector of Katz centralities \mathbf{x} and the approximations $\widehat{\mathbf{x}}$ computed with [Algorithms 6.1](#) and [6.2](#). We also look at the *intersection similarity* of the induced rankings [19]. The intersection similarity of two ranking vectors β and γ is defined as

$$isim_p(\beta, \gamma) = \frac{1}{p} \sum_{i=1}^p \frac{|\beta_{1:i} \Delta \gamma_{1:i}|}{2i},$$

where, using Matlab notation, we denote by $\beta_{1:i}$ and $\gamma_{1:i}$ the vectors of length i containing the first i elements in β and γ , respectively, and we denote by Δ the symmetric difference operator. The intersection similarity is a popular distance often used to compare the rankings given by two centrality measures; see, e.g., [3, 6, 12]. If $isim(\beta, \gamma) \approx 0$, then the two rankings are almost identical, whereas $isim(\beta, \gamma) = 1$ represents two completely different rankings.

[Figures 3](#) and [4](#) plot the evolution of the 2-norm error (top) and of the intersection similarity between the ranking vectors induced by the ground truth \mathbf{x} and by the computed approximations $\widehat{\mathbf{x}}$ with $p = \lceil n/100 \rceil$ (bottom) as a function of the percentage of nodes/edges being removed. The relative error stabilizes at around 10^{-2} . The intersection similarity also stabilizes at very low values, indicating that the approximations computed using [Algorithms 6.1](#) and [6.2](#) correctly identify the nodes ranked in the top 1% by exact Katz centrality after each removal.

We now consider the same experiment, but instead of removing elements at random, we remove the top 1% of the highest ranked nodes (according to Katz centrality). It is well known that removing these elements will significantly affect the centrality scores and the induced rankings; see, e.g., [36]. We plot the results of our experiment in [Figure 5](#). These show that our approximations retain a similar error rate even though the centralities changed significantly from their initial value. The values of intersection similarity for these experiments are larger than in the previous one, but still very low. This confirms that our strategies well reproduce both scores and rankings, regardless of node removal strategy.

The last set of experiments examines the behaviour of the bounds in [Theorems 5.1](#) and [5.2](#) when 1% of nodes or edges selected uniformly at random are removed from the two real-world networks in our dataset. These tests mimic scenarios where a random

Updating Katz centrality by counting walks

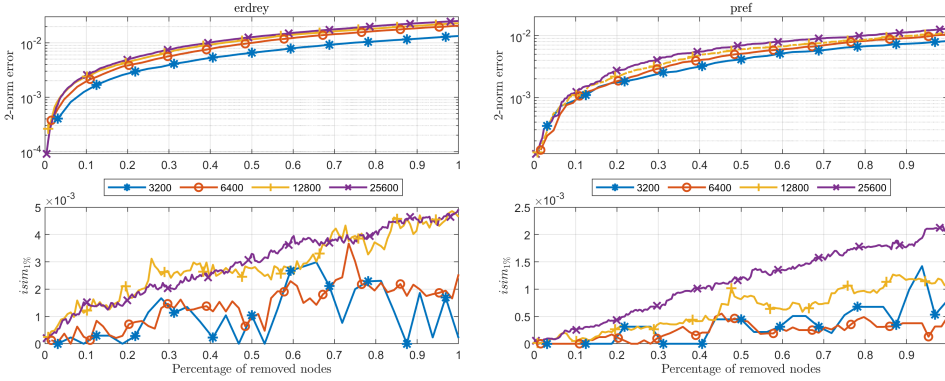


FIGURE 3. *Random removal of a sequence of nodes. Evolution of the relative 2-norm error and $isimp_p$, where $p = \lceil n/100 \rceil$, when sequentially removing nodes from synthetic networks of increasing sizes.*

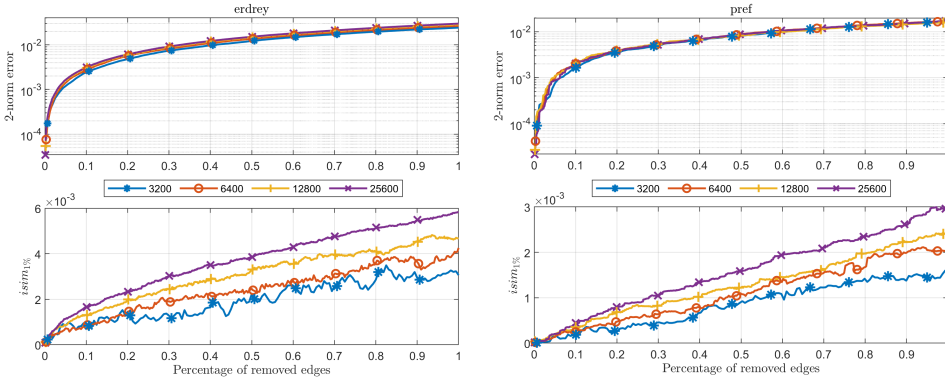


FIGURE 4. *Random removal of a sequence of edges. Evolution of the relative 2-norm error and $isimp_p$, where $p = \lceil m/100 \rceil$, when sequentially removing edges from synthetic networks of increasing sizes.*

event shuts down roads/intersections in the Minnesota road network, or isolates a group of routers within the network AS-735 of autonomous systems. As before, the results are averaged over 30 runs.

The results are displayed in Figures 6 and 7, where we plot the change in the total network communicability as nodes (resp., edges) are sequentially removed from the network. On the same plots, we display the upper bounds obtained in (5.1) and (5.3). The bounds are extremely close to the actual values of total network communicability. A remark is however in order; the sharpness of these bounds strongly depends on the change in the underlying centrality scores. If these change significantly, then using the previous value of Katz centrality to obtain an upper bound will, over time, cause a severe overestimation of the change in total network communicability. This situation may arise when some crucial elements are removed from the network or when the number of removed elements becomes large. In these situations, one could mitigate

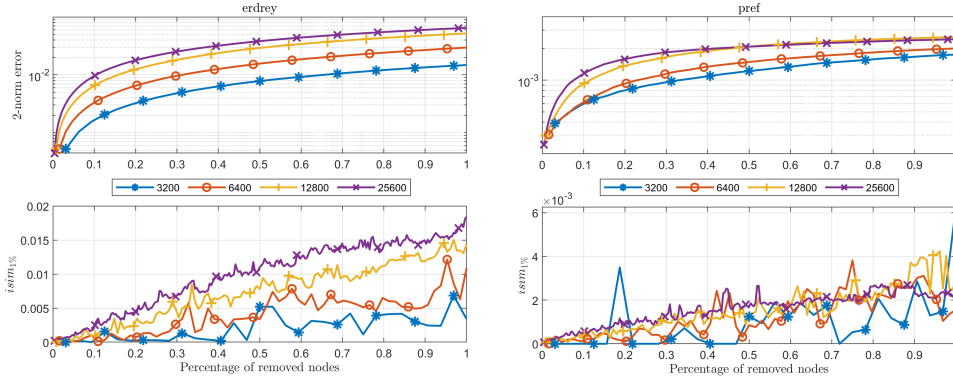


FIGURE 5. Targeted removal of a sequence of nodes. Evolution of the relative 2-norm error and $isimp$, where $p = \lfloor n/100 \rfloor$, when sequentially removing 1% of the most important nodes (according to Katz centrality) from synthetic networks of increasing sizes. All the values are averaged over 30 runs.

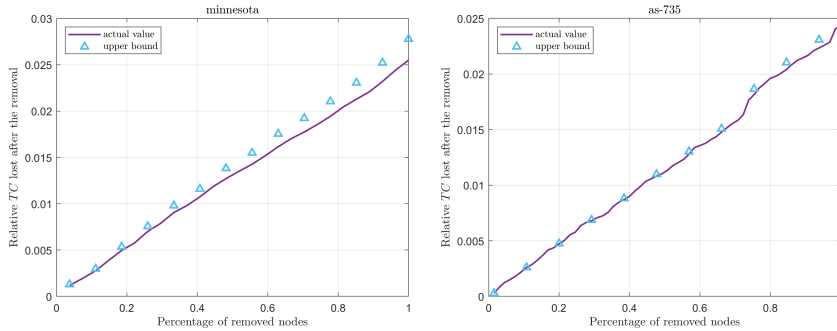


FIGURE 6. Comparison between the bound (5.1) and the actual value of TC lost (relative to the initial value) after the removal of 1% of the total nodes from the Minnesota roads the as-735 connections networks, taken uniformly at random. The values are averaged over 30 runs.

the error by selectively recomputing the measures after a suitable number of iterations; see, e.g., [10] for some strategies.

8. Conclusions and future work. In this paper we derived explicit formulas, expressed in terms of loss of walks, for the change in Katz centrality when nodes and/or edges are removed from a simple graph. These formulas informed the description of two algorithms to approximate the new centrality vectors after network modification. The theoretical computational cost of the proposed methods is $\mathcal{O}(Lm)$, where m is the number of undirected edges in the original graph and L is a (dynamically chosen) maximum walk-length. We also described upper bounds on the change in total network communicability. Numerical results on both synthetic and real-world networks showed that our approximation strategies are more efficient than others available in the literature, including recomputing from scratch and series truncation, and return results that are accurate both in terms of numerics (relative error) and ranking (intersection similarity).

Given these promising results, future work will focus on developing similar strate-

Updating Katz centrality by counting walks

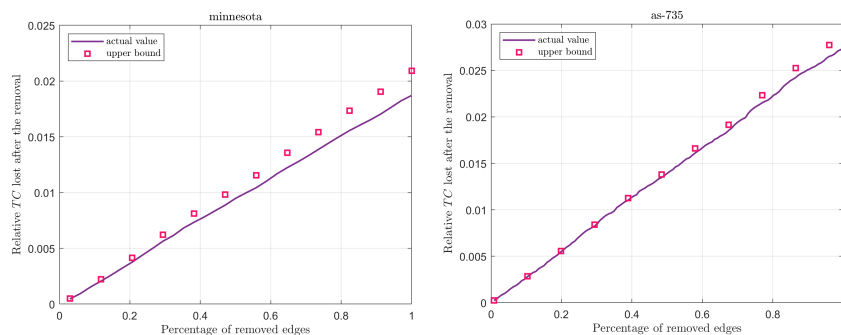


FIGURE 7. Comparison between the bound (5.3) and the actual value of TC lost (relative to the initial value) after the removal of 1% of the total edges from the Minnesota roads and the as-735 connections networks, taken uniformly at random. The values are averaged over 30 runs.

gies for other graph types and centrality measures.

Acknowledgements. We would like to thank the referees for their insightful comments, that helped us improve the presentation of the paper.

REFERENCES

- [1] R. ALBERT, I. ALBERT, AND G. L. NAKARADO, *Structural vulnerability of the north american power grid*, Phys. Rev. E, 69 (2004), p. 025103.
- [2] F. ARRIGO AND M. BENZI, *Updating and downdating techniques for optimizing network communicability*, SIAM Journal on Scientific Computing, 38 (2016), pp. B25–B49.
- [3] F. ARRIGO, A. GAUTIER, AND F. TUDISCO, *Node and layer eigenvector centralities for multiplex networks*, SIAM Journal on Applied Mathematics, 78 (2018), pp. 853–876.
- [4] B. BECKERMANN, D. KRESSNER, AND M. SCHWEITZER, *Low-rank updates of matrix functions*, SIAM Journal on Matrix Analysis and Applications, 39 (2018), p. 539–565.
- [5] M. BENZI AND P. BOITO, *Matrix functions in network analysis*, GAMM-Mitteilungen, 43 (2020).
- [6] M. BENZI AND C. KLYMKO, *Total communicability as a centrality measure*, Journal of Complex Networks, 1 (2013), pp. 124–149.
- [7] ———, *On the limiting behavior of parameter-dependent network centrality measures*, SIAM Journal on Matrix Analysis and Applications, 36 (2015), pp. 686–706.
- [8] D. S. BERNSTEIN AND C. F. VAN LOAN, *Rational matrix functions and rank-1 updates*, SIAM Journal on Matrix Analysis and Applications, 22 (2000), pp. 145–154.
- [9] D. BERTACCINI AND F. DURASTANTE, *Iterative methods and preconditioning for large and sparse linear systems with applications*, Monographs and Research Notes in Mathematics, CRC Press, Boca Raton, FL, 2018.
- [10] D. BERTACCINI AND A. FILIPPO, *A proposal for ranking through selective computation of centrality measures*, PLOS ONE, 18 (2023), pp. 1–16.
- [11] H. CHAN, L. AKOGLU, AND H. TONG, *Make it or break it: Manipulating robustness in large networks*, in Proceedings of the SIAM International Conference on Data Mining, United States, 2014, SIAM, pp. 325–333.
- [12] S. CIPOLLA, F. DURASTANTE, AND F. TUDISCO, *Nonlocal PageRank*, ESAIM: M2AN, 55 (2021), pp. 77–97.
- [13] R. COHEN, K. EREZ, D. BEN AVRAHAM, AND S. HAVLIN, *Breakdown of the internet under intentional attack*, Phys. Rev. Lett., 86 (2001), pp. 3682–3685.
- [14] T. DAVIS, W. HAGER, AND I. DUFF, *SuiteSparse*, http://faculty.cse.tamu.edu/davis/suite_sparse.html, (2014).
- [15] T. A. DAVIS, *Direct Methods for Sparse Linear Systems*, SIAM, 2006.
- [16] O. DE LA CRUZ CABRERA, J. JIN, S. NOSCHESSE, AND L. REICHEL, *Communication in complex networks*, Applied Numerical Mathematics, 172 (2022), pp. 186–205.
- [17] J. A. DUNNE, R. J. WILLIAMS, AND N. D. MARTINEZ, *Network structure and biodiversity loss in food webs: robustness increases with connectance*, Ecology Letters, 5 (2002), pp. 558–567.

- [18] E. ESTRADA, *The Structure of Complex Networks: Theory and Applications*, Oxford University Press, 2011.
- [19] R. FAGIN, R. KUMAR, AND D. SIVAKUMAR, *Comparing top k lists*, SIAM J. Discret. Math., 17 (2003), pp. 134–160.
- [20] K. FOSTER, S. MUTH, J. POTTERAT, AND R. ROTHENBERG, *A faster Katz status score algorithm*, Computational & Mathematical Organization Theory, 7 (2001), pp. 275–285.
- [21] P. GRINDROD, D. J. HIGHAM, AND V. NOFERINI, *The deformed graph Laplacian and its applications to network centrality analysis*, SIAM Journal on Matrix Analysis and Applications, 39 (2018), pp. 310–341.
- [22] P. GRINDROD, M. C. PARSONS, D. J. HIGHAM, AND E. ESTRADA, *Communicability across evolving networks*, Phys. Rev. E, 83 (2011), p. 046120.
- [23] D. J. HIGHAM AND A. TAYLOR, *CONTEST: A controllable test matrix toolbox for MATLAB*, ACM Trans. Math. Softw., 35 (2009).
- [24] N. HIGHAM, *Functions of Matrices: Theory and Computation*, SIAM, 2008.
- [25] P. HOLME, B. J. KIM, C. N. YOON, AND S. K. HAN, *Attack vulnerability of complex networks*, Phys. Rev. E, 65 (2002), p. 056109.
- [26] R. A. HORN AND C. R. JOHNSON, *Matrix Analysis*, Cambridge university press, 2012.
- [27] S. IYER, T. KILLINGBACK, B. SUNDARAM, AND Z. WANG, *Attack robustness and centrality of complex networks*, PLOS ONE, 8 (2013), pp. 1–17.
- [28] B. JACKSON AND A. D. SOKAL, *Maxmaxflow and counting subgraphs*, Electron. J. Comb., 17 (2007).
- [29] L. KATZ, *A new status index derived from sociometric analysis*, Psychometrika, 18 (1953), pp. 39–43.
- [30] A. N. LANGVILLE AND C. D. MEYER, *Updating Markov chains with an eye on Google’s Page-Rank*, SIAM Journal on Matrix Analysis and Applications, 27 (2006), pp. 968–987.
- [31] W. LI, M. CHU, AND J. QIAO, *A pruning feedforward small-world neural network based on Katz centrality for nonlinear system modeling*, Neural Networks, 130 (2020), pp. 269–285.
- [32] E. NATHAN AND D. BADER, *Incrementally updating Katz centrality in dynamic graphs*, Social Network Analysis and Mining, 8 (2018), p. 26.
- [33] E. NATHAN AND D. A. BADER, *Approximating personalized Katz centrality in dynamic graphs*, in Parallel Processing and Applied Mathematics, Cham, 2018, Springer, pp. 290–302.
- [34] M. E. J. NEWMAN, *Networks: An Introduction*, Oxford University Press, 2010.
- [35] V. NOFERINI AND R. WOOD, *Efficient computation of Katz centrality for very dense networks via negative parameter Katz*, Journal of Complex Networks, 12 (2024), p. cnae036.
- [36] S. POZZA AND F. TUDISCO, *On the stability of network indices defined by means of matrix functions*, SIAM Journal on Matrix Analysis and Applications, 39 (2018), pp. 1521–1546.
- [37] Y. SAAD, *Iterative Methods for Sparse Linear Systems*, SIAM, 2003.
- [38] Y. SATOTANI, T. MIGITA, AND N. TAKAHASHI, *An algorithm for updating betweenness centrality scores of all vertices in a graph upon deletion of a single edge*, Journal of Complex Networks, 10 (2022), p. cnac033.
- [39] M. SCHWEITZER, *Sensitivity of matrix function based network communicability measures: computational methods and a priori bounds*, SIAM Journal on Matrix Analysis and Applications, 44 (2023), pp. 1321–1348.
- [40] J. SHERMAN AND W. J. MORRISON, *Adjustment of an inverse matrix corresponding to a change in one element of a given matrix*, The Annals of Mathematical Statistics, 21 (1950), pp. 124–127.
- [41] J. ZHAN, S. GURUNG, AND S. P. K. PARSA, *Identification of top-k nodes in large networks using Katz centrality*, Journal of Big Data, 4 (2017), p. 16.
- [42] J. ZHAO, T.-H. YANG, Y. HUANG, AND P. HOLME, *Ranking candidate disease genes from gene expression and protein interaction: a Katz-centrality based approach*, PLOS ONE, 6 (2011), p. e24306.