

NEW IDEAS FOR WALK-BASED MEASURES IN COMPLEX NETWORKS

PhD in
Matematics

PhD Cycle
XXXVIII

Author
Alessandro Filippo

Supervisor
Prof. Daniele Bertaccini

Coordinator
Prof. Martina Lanini

A.Y. 2025/2026

Abstract

Many complex systems that permeate our lives, such as the Internet, urban transportation systems, and the electrical power grid, can be represented as networks. Analysing their structural properties can help us better understand the systems they represent.

Centrality and communicability measures address two fundamental issues in network analysis. Centrality indices are used to identify the most important nodes in a network, while communicability measures quantify how effectively two nodes can exchange information with each other.

In this thesis, we develop new tools for a well-known class of centrality and communicability measures based on walk combinatorics that can be expressed in terms of entries (or sums of entries) of a matrix function $f(A)$, where A is the network adjacency matrix.

The original contributions of this thesis are organised as follows. First, we address the problem of updating Katz centrality after the removal of a subset of nodes or edges from a simple network. Next, we establish upper and lower bounds on the change in f-communicability and related network indices caused by perturbing (adding, deleting, or modifying the weights of) a set of edges in a directed and possibly weighted network. Finally, we introduce a new walk-based centrality measure that is efficient to compute and specifically tailored to the analysis of certain biological networks that can be modelled as directed acyclic graphs.

Selected numerical experiments on both synthetic and real-world networks validate our findings for all topics considered.

Acknowledgements

First, I am thankful to my PhD advisor, Prof. Daniele Bertaccini, for providing guidance and advice throughout these years, as well as for introducing me to the topics that would later become the object of my PhD studies.

I am grateful to Dr. Francesca Arrigo for her support and for everything she has patiently taught me since my first visit to the University of Strathclyde back in November 2023.

I would like to extend my sincere thanks to Prof. Vanni Noferini for the hospitality and financial support he provided during my research visit to Aalto University in September 2025.

I would also like to thank my co-author Luigi Chiricosta for introducing me to the world of biological networks and for our fruitful collaboration.

I also want to acknowledge Prof. Carla Manni, Prof. Carmine Di Fiore, Dr. Mariarosa Mazza, and my PhD colleagues Emilia Cozzolino and Simone Marrocco for the countless advice I received during my time at Tor Vergata.

Roma, March 2026,

Alessandro Filippo

Contents

Introduction **x**

1 Background **1**
1.1 Basics of graph theory 1
1.2 Few topics of linear algebra 3
1.2.1 Matrix functions 4
1.3 Few topics of network science 7
1.3.1 Centrality measures 7
1.3.2 Communicability measures 9

2 Updating Katz centrality in simple graphs **12**
2.1 Problem setting 12
2.2 Walk counting in a simple graph 13
2.2.1 Counting walks through a set to edges 14
2.2.2 Counting walks through a set of nodes 15
2.3 Main results 22
2.3.1 Updating Katz centrality via walk count 23
2.3.2 Bounds on total network communicability 26
2.4 Algorithms 28
2.4.1 Removing nodes 28
2.4.2 Removing edges 30
2.4.3 Computational cost 31
2.5 Numerical experiments 31
2.6 Conclusions 36

3 Estimating f-communicability in digraphs **40**
3.1 Problem setting and notation 40
3.2 A special class of communicability measures 43
3.3 Counting walks through a set of directed edges 46
3.4 Main results 48

3.4.1 Estimating the change in f-communicability 49
3.4.2 Bounds on related network indices 51
3.5 Extension to weighted graphs 52
3.6 Numerical examples 56
3.7 Conclusions and outlook 59

4 A new centrality measure for acyclic digraphs 62
4.1 Some centrality measures for DAGs 62
4.2 Walk counting in a DAG 63
4.3 A new centrality score for DAGs 65
4.3.1 Source-to-sink communicability, stsc 65
4.3.2 Computing stsc 67
4.3.3 Update 68
4.4 Numerical experiments 71
4.4.1 Tests on toy DAGs 71
4.4.2 Tests on larger DAGs 74
4.5 Application to protein interaction networks 75
4.5.1 Protein interaction networks 77
4.5.2 Pathways description 78
4.5.3 Pathways preprocessing 78
4.5.4 Analysis and comparisons 79
4.6 Conclusions 84

5 Conclusions and future work 85

Bibliography 87

List of Figures

1 Wikipedia pages network in November 2005. Figure from [36] . xi
2 US Airport network. Here nodes are coloured according to their degree (number of connections). Data taken from Gephi [16]. . xii

2.1 Toy network. 18
2.2 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 2.2.2. 19
2.3 Random removal of a sequence of nodes (top figure) or edges (bottom figure). Evolution of the relative 2-norm error and $isim_p$, where $p = \lceil m/100 \rceil$, when sequentially removing edges from synthetic networks of increasing sizes. 37
2.4 Targeted removal of a sequence of nodes. Evolution of the relative 2-norm error and $isim_p$, where $p = \lceil n/100 \rceil$, 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. 38
2.5 Comparison between the bound (2.14)–(2.16) and the actual value of TC lost (relative to the initial value) after the removal of 1% of the total nodes (top figure) or edges (bottom figure) form the `minnesota` and the `as-735` networks, taken uniformly at random. The values are averaged over 30 runs. 39

3.1 Update and downdate problem on a toy graph. 42
3.2 Weight perturbation problem on a toy graph with four nodes. . 53
3.3 A synthetic graph that consists in two undirected cycles, with eleven nodes each, connected by a direct edge between node 11 and 12. In our experiment, we consider the case in which the virtual edge $e = (12, 11)$, represented by a dashed line in the figure, is added to the graph. 57

3.4 Comparisons between the bounds (3.5), the upper bound in [91] with $\epsilon = 0.03$ for $r_{1/3}(z)$ (no parameters are needed for $\exp(z)$), and the actual value of $f(M_{\mathcal{E}})_{ii} - f(M)_{ii}$ after the addition of the edge $e = (12, 11)$ to the graph in Figure 3.3. In the plots, f is chosen to be either $f(z) = \exp(z)$ (top figure) or $f(z) = r_{1/3}(z) = (1 - z/3)^{-1}$ with (bottom figure). 58

3.5 Bounds (3.6)–(3.7) compared with the actual value of $\mathrm{EE}_f(M_{\mathcal{E}}) - \mathrm{EE}_f(M)$ and $\mathrm{TC}_f(M_{\mathcal{E}}) - \mathrm{TC}_f(M)$ after the sequential removal of 15 edges (chosen uniformly at random) from the `gnutella08` network. Here f is chosen to be either $f(z) = \exp(z)$ (top right and bottom right figures) or $f(z) = r_\alpha(z) = (1 - \alpha z)^{-1}$ with $\alpha = 0.85/\rho(A)$ (top left and bottom left figures). All the values are averaged over 20 runs with the same parameters. 60

3.6 Comparisons between the bounds (3.9)–(3.10) and the actual variation of $\mathrm{EE}_f(A)$ and $\mathrm{TC}_f(A)$ after the addition of $\Delta\omega$ to the weight of an existing edge (chosen uniformly at random) of the `sandi-authors` network. Here, f is chosen to be either $f(z) = \exp(z)$ (top right and bottom right figures) or $f(z) = r_\alpha(z) = (1 - \alpha z)^{-1}$, with $\alpha = 0.85/\rho(A)$ (top left and bottom left figures). All the values are averaged over 20 runs with the same parameters. 61

4.1 Toy network from Example 1 72

4.2 Toy network from Example 2 73

4.3 Toy network form Example 3 74

4.4 Timings (in seconds) to compute the *stsc* scores for some synthetic DAGs with increasing sizes after the removal/failure of a randomly chosen edge. The two dashed lines represent, respectively, the results obtained with the computational scheme described in Section 4.3.2 in the case of sorted and unsorted DAGs. The continuos line shows the results obtained with the update formula presented in Section 4.3.3. The latter is tested only in the case of topologically sorted DAGs. All the values are based on MATLAB's Gaussian elimination and are averaged over 50 runs with the same parameters. Error bars represent standard deviation 76

List of Tables

2.1 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. 32

2.2 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 $\alpha = 0.85/\rho(A)$. 32

2.3 Number of iterations and CPU timings required for approximating $\mathbf{x}^{\mathcal{S}}$ with methods (i)–(iii) after the removal of one node selected uniformly at random. The chosen tolerances are `tol`= 10^{-5} for method (i), and `tol`= 10^{-4} for methods (ii)–(iii). The initial guess for `pcg` is specified in brackets. All the values are averaged over 30 runs. 34

2.4 Number of iterations and CPU timings required for approximating $\mathbf{x}^{\mathcal{S}}$ with methods (i)–(iii) after the removal of one edge selected uniformly at random. The chosen tolerances are `tol`= 10^{-5} for method (i), and `tol`= 10^{-4} for methods (ii)–(iii). The initial guess for `pcg` is specified in brackets. All the values are averaged over 30 runs. 35

3.1 A scheme summarising the notation introduced in Section 3.1. Here, the symbols ✓and **?** indicate whether a given quantity is assumed to be known in a particular problem or is, instead, the objective of our analysis. 43

3.2 Some examples of functions in $\mathcal{P}_\gamma$ for some $\gamma \in [0, 1]$ and some examples of functions in $\mathcal{P}$ that do not belong to $\mathcal{P}_\gamma$ for any $\gamma \in [0, 1]$. 46

4.1 Centrality scores from Example 1 72

4.2 Centrality scores from Example 2 73

4.3 Centrality scores from Example 3 75
4.4 Description of the networks obtained from the pathway preprocessing 79
4.5 Kendall correlation coefficient between the rankings given by *stsc* and those given by the other centrality measures described in Section 4.4 80
4.6 Top 10 nodes of the `hsa05022` network ranked using *stsc* and the other centrality measures described in Section 4.4 81
4.7 Top 10 nodes of the `hsa05200` network ranked using *stsc* and the other centrality measures described in Section 4.4 82
4.8 Top 10 nodes of the `hsa05010` network ranked using *stsc* and the other centrality measures described in Section 4.4 83
4.9 Top 10 nodes of the `hsa04010` network ranked using *stsc* and the other centrality measures described in Section 4.4 84

Introduction

Complex networks can be used to model interconnected systems across all realms of life, from biological systems, such as food webs [41] or protein interaction networks [14], to human-made structures such as the World Wide Web [76], social networks [101], or the global airline transportation networks [59], just to name a few. Because of the wide range of applications, Network Science, the field concerned with the study of complex networks, is now a well established (and highly interdisciplinary) field [15].

Mathematically, a complex network can be modelled as a graph [63], in which the nodes represent the various components of the system and the edges represent the relations that connect them [44]. Complex networks arising from real-life applications usually have a power-law degree distribution, a non-zero degree-degree correlation, a large number of connected triangles, and some form of community structure [86]. Thus, they significantly differ from standard examples studied in classical graph theory, as they are not "regular" nor similar to a Erdős–Rényi random graph [43].

Modern real-world networks also tend to be extremely large in size, with hundreds of thousands (sometimes millions) nodes and edges. In such large-scale settings, networks are hard to visualise (see, e.g., the Wikipedia pages network shown in Figure 1), and often it is hard to analyse their structural properties without using efficient methods borrowed from numerical linear algebra. Central to this approach is the fact that each graph with n nodes can be associated with (and studied via operations involving) an $n \times n$ adjacency matrix A, which can be very large in size but also very sparse if the number of edges is not greater than $O(n)$.

As a further degree of complexity, these networks are rarely static, as nodes and edges can appear or disappear over time for various reasons. This is the case of a social media platform where users join or leave at any given moment, or of the continually expanding World Wide Web; keeping track of changes in their structural properties (such as community structure) can require substantial computational resources.

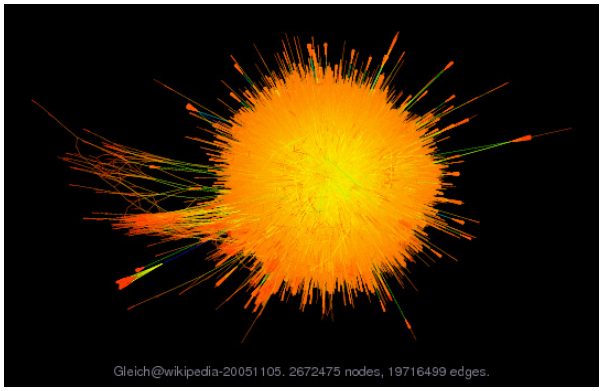


Figure 1: Wikipedia pages network in November 2005. Figure from [36]

A typical task in complex network analysis is determining which nodes are the most "important" within a network. For example, identifying the most popular or influential people in a social network can help to understand how news or infections flow through it, while identifying the most vulnerable parts of an infrastructural network can help defend it from random failures or targeted attacks (see, e.g., the US airport network in Figure 2). For this reason, reliable metrics for ranking nodes are extremely desirable in practical applications, which led to the introduction of several centrality measures. A node centrality measure is defined as a function $c : V \longrightarrow \mathbb{R}_{\geq 0}$. The larger the score, the more important node $i \in V$ is within the network. The most popular examples of centrality measures are degree centrality, betweenness centrality, closeness centrality [52], and eigenvector centrality [28]. However, many other measures have been introduced over the years, often designed with a specific application in mind; see for example [44, Chapter 7] and [87, Chapter 7].

Another common task in network design and optimization is to quantify how effectively various components of a network, such as individual nodes or groups of nodes, communicate with each other. Historically, communication between nodes was considered to occur exclusively along geodetic paths [48]. However, the realisation that "information" in a network can propagate along all possible routes and not just the shortest ones (long traversals are simply less likely), led to a more general definition of communicability between nodes [45].

This thesis focuses on a widely used class of centrality and communicability measures based on walks and, more precisely, on their combinatorics. Centrality measures in this class assign to each node a nonnegative score based on

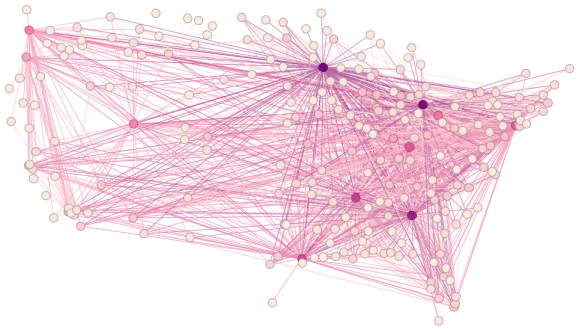


Figure 2: US Airport network. Here nodes are coloured according to their degree (number of connections). Data taken from Gephi [16].

the number of walks that start from it, while communicability measures in this class do the same for each pair of nodes by counting all walks that connects them. In both cases, walks are weighted by an attenuation factor that depends on their length, which is chosen so that we can consider walks up to an infinite length [47]. The fundamental principle underlying this class of measures (and that is paramount for their fast computation) is the walk-counting property of the adjacency matrix A (see, e.g., [99]), which states that the number of walks of length r connecting nodes i and j is equal to the (i, j)th entry of A^r. This property allows us to express all centrality and communicability measures in this class as entries (or sums of thereof) of a suitable matrix function

$$f(A) = \sum_{r=0}^{\infty} c_r A^r.$$

By doing so, we are essentially translating a combinatorial problem (counting walks in a graph) into a potentially much more efficient numerical algorithm: evaluating a matrix function. More precisely, consider a sequence of positive real numbers $\{c_r\}_{r\geq 0}$ such that $\sum_{r=0}^{\infty} c_r A^r$ convergences to the matrix function $f(A)$. We can define [47]:

- the f-centrality of node i as $f(A)_{ii}$;
- the f-communicability between node i and node j as $f(A)_{ij}$; and
- the f-total communicability of node i as $[f(A)\mathbf{1}]_i$,

where $\mathbf{1}$ denotes the vector of all ones. Notice that, if the coefficients c_r are chosen such that $c_r = 1/k!$ or $c_r = \alpha^r$ for all $r \geq 0$, with $0 < \alpha\rho(A) < 1$, we obtain the following well-known measures:

- Subgraph centrality [49]: $\sum_{r=0}^{\infty} \frac{1}{r!}(A^r)_{ii} = \exp(A)_{ii}$;
- Subgraph communicability [45]: $\sum_{r=0}^{\infty} \frac{1}{r!}(A^r)_{ij} = \exp(A)_{ij}$; and
- Katz centrality [73]: $\sum_{r=0}^{\infty} \alpha^r [A^r \mathbf{1}]_i = [(I - \alpha A)^{-1}\mathbf{1}]_i$.

Walk-based measures offer several advantages that make them appealing for the analysis of modern real-world networks. First, their interpretation in terms of matrix functions allows them to be computed using efficient numerical linear algebra tools such as Krylov subspace methods (see, for example, [19] for some techniques). This feature makes walk-based measures much faster to compute than shortest-path-based measures such as betweenness and closeness centrality, particularly in the case of large networks. Moreover, their ability to account for all walks rather than just shortest paths makes them suitable for applications where information does not exclusively travel along shortest routes. Furthermore, they can be easily generalised or specialised for a specific type of network. For instance, one may choose to exclude [57] (or simply downweight [10]) certain types of walks from the count, or consider temporal walks in temporal networks [11, 58, 88], or do both. They can also be used to detect hubs and authorities within a directed network [20]. However, to the best of our knowledge, prior to [24], no walk-based centrality measure specifically exploited the structure of a directed acyclic network.

Despite their computational advantages, walk-based measures are hard to keep up-to-date if the networks are large and/or if they undergo frequent structural changes. Consider, for instance, the opening of a new route in a public transportation network or the disconnection of a power plant from a power grid caused by an unexpected breakdown. When centrality or communicability are of interest in these and similar situations, doing full recomputation of the measures after every modification may not be feasible if the networks are large, and approximating strategies may be devised instead. As a consequence, numerous studies have recently investigated how graph perturbations, such as the removal or the addition of a subset of edges to the graph, affect walk-based measures [5, 6, 17, 38, 91, 95]. This issue may be particularly relevant in real-life applications, in which simulating the removal of edges is often used as a way to address issues related to the robustness of the network or the presence of communication bottlenecks (see, e.g., [2, 31, 34, 42, 66, 69]).

In this thesis, we consider these lines of investigation by developing new theoretical and computational tools for walk-based centrality and communicability measures. Our approach integrates techniques from graph theory, numerical analysis, linear algebra, and (especially) combinatorics. In fact, central to our work is the derivation of several combinatorial results concerning the enumeration of walks in a graph that visits a specific set of nodes or edges. We apply these results to estimate Katz centrality and f-communicability in Chapters 2 and 3, and to define a new centrality measure for directed acyclic networks in Chapter 4. More in detail, the thesis is structured as follows:

- In Chapter 1 we recall some background definitions and results from graph theory and linear algebra that are essential for this thesis. We also review some popular centrality and communicability measures used in networks science.

- In Chapter 2 we derive explicit formulas, expressed in terms of loss of walks, for the change in the Katz centrality after the removal of a subset of edges or nodes from a simple graph, i.e., the change in the vector

$$\mathbf{x} = \sum_{r=0}^{\infty} \alpha^r A^r \mathbf{1} = (I - \alpha A)^{-1} \mathbf{1},$$

  where $A \in \mathbb{R}^{n \times n}$ is the adjacency matrix of the graph and $\alpha > 0$ is such that $0 < \alpha\rho(A) < 1$. This chapter is based on

  [7] F. Arrigo, D. Bertaccini, and A.F. *Updating Katz centrality by counting walks.* SIAM Journal on Matrix Analysis and Applications, 2025.

- In Chapter 3 we introduce a special class of analytic functions:

$$\mathcal{P}_\gamma = \{f \in \mathcal{P} : f(z) = \sum_{p=0}^{\infty} c_p z^p, \\ \text{with } \gamma^{p+q} c_p c_q \leq c_{p+q} \leq c_p c_q \text{for all } p, q \geq 0\},$$

  where $\gamma \in [0, 1]$, and $\mathcal{P}$ denotes the class of analytic functions with (finite or infinite) radius of convergence and strictly positive Taylor coefficients [22]. We establish bounds for the change in f-communicability resulting from the perturbation (addition, deletion, or weight modification) of a set of edges in a graph, where $f \in \mathcal{P}_\gamma$. This chapter is based on

[9] F. Arrigo and A. F. *Estimating f-communicability in digraphs after edge perturbation*, in preparation.

- In Chapter 4 we introduce *source-to-sink communicability*, a novel walk-based centrality measure for directed acyclic networks such as biological pathways [78] or articles citation networks [39], and we show that it can be derived from the solutions of two linear systems of the form

$$(I - M)\mathbf{x} = \mathbf{b},$$

where $M \in \mathbb{R}^{n\times n}$ is similar to a strictly upper triangular (up to a permutation) and $\mathbf{b} \in \{0,1\}^{n\times n}$. We also present a method to update the newly introduced measure after the removal of an edge from the graph. This chapter is based on

[24] D. Bertaccini, L. Chiricosta, and A. F. *Source-to-sink communicability: a new centrality measure for directed acyclic networks.* Numerical Algorithms, 2025.

- Finally, Chapter 5 contains concluding remarks and an outlook on potential directions for future work.

1. Background

This chapter outlines basic definitions and notation from graph theory, linear algebra, and network science that are essential for this thesis. For details on the formulation of such topics we refer for instance to [30, 44, 65].

In the following, I indicates the identity matrix while $\mathbf{e}_i$ represents its ith column. Moreover, $\mathbf{1}$ and $\mathbf{0}$ denote, respectively, the vectors of all ones and all zeros.

1.1. Basics of graph theory

A *graph* or *network* is a pair $G = (V, E)$, where $V = \{i : i = 1, \dots, n\}$ is the set of *nodes* (or *vertices*) and $E \subseteq V \times V$ is the collection of *edges* (or *arcs*) that connects them. We say that a graph G is *undirected* if E is symmetric; otherwise we say that G is *directed* (or a *digraph*).

Definition 1.1 (simple graph)**.** *An undirected graph with no self-loops, i.e., edges of the form (i, i) nor repeated edges is said to be* simple*.*

Definition 1.2 (topologically sorted graph)**.** *A graph is said to be* topologically sorted *if $(i, j) \in E$ implies $i < j$.*

An edge $(i, j) \in E$ is said to be *reciprocated* if $(j, i) \in E$ as well. In an undirected graph, every edge is reciprocated and we simply write $\{i, j\}$ to denote the edge between i and j. A pair $(i, j) \in \overline{E}$, where $i \neq j$ and $\overline{E}$ denotes the complement of E, is usually called *virtual edge*.

Definition 1.3 (degree)**.** *Two nodes i, j are said to be* adjacent*, and we write $i \sim j$, if there exists an edge between them. For each $i \in V$, we define its* degree $\deg(i) = |\{v \in V : i \sim v\}|$ *as the number of nodes adjacent to it.*

Definition 1.4 (indegree and outdegree)**.** *For each $i \in V$, we define its* indegree $\mathrm{ind}(i) = |\{(v, i) \in E : v \in V\}|$ *as the number of edges ending in i and its* out-degree $\mathrm{outd}(i) = |\{(i, v) \in E : v \in V\}|$ *as the number of edges starting from it. If G is undirected, then* $\mathrm{ind}(i) = \mathrm{outd}(i) = \deg(i)$*.*

Definition 1.5 (source and sink)**.** *A node $i \in V$ such that* $\mathrm{ind}(i) = 0$ *is called a* source*, while a node with* $\mathrm{outd}(i) = 0$ *is called a* sink*. If* $\mathrm{ind}(i) = \mathrm{outd}(i) = 0$*, we say that i is* isolated*.*

Definition 1.6 (r-walk)**.** *A* walk *of length $r > 0$, or* r-walk*, from a node i to a node j is a list of $r + 1$ nodes $i_0 = i, i_1, \ldots, i_r = j$ such that $(i_k, i_{k+1}) \in E$ for each $k = 1, \ldots, r$. A single node is also a walk of length $r = 0$.*

An r-walk in which $i_0 = i_r = i$ is said to be *closed*. A walk with no repeated nodes is called *path*, and a closed path is called *cycle*.

Definition 1.7 (directed acyclic graph)**.** *A directed graph with no cycles is said to be* acyclic*, or a DAG for short.*

The *diameter* of a graph is the maximum path length and the *(combinatorial) distance* $\mathrm{dist}(i, j)$ between two nodes i and j is the length of the *shortest (geodetic) path* that connects them. If such a path does not exist, we set $\mathrm{dist}(i, j) = +\infty$. If $\mathrm{dist}(i, j) < +\infty$ for each pair of nodes i, j, then we say that G is *strongly connected*, if this property holds only when the orientation of the edges is disregarded we say that G is just *connected*. A strongly connected subgraph $S = (V', E')$, with $V' \subseteq V$ and $E' \subseteq E$, is a *strongly connected component* of G.

Definition 1.8 (virtual walk)**.** *A* virtual walk *of length r in G is a list of $r + 1$ nodes $i_0, i_1, \ldots, i_r$, with $i_k \neq i_{k+1}$*[1] *for each $k = 0, \ldots, r$, such that $(i_k, i_{k+1}) \in \overline{E}$ for at least one $k = 0, \ldots, r$.*

Let $e = (u, v)$ be an edge (resp., virtual edge) in G, we say that a walk (resp., virtual walk) $i_0, i_1, \ldots, i_r$ *visits* e if there exists at least one $k = 0, \ldots, r$ such that $i_k = u$ and $i_{k+1} = v$. If a walk (resp., virtual walk) $i_0, i_1, \ldots, i_r$ is such that $i_k = w$ for some $k = 0, \ldots, r$, we say that *visits* $w \in V$. Since, by definition, no walk in G can visit any virtual edge, we will henceforth omit the "virtual" specification when referring to virtual walks that visit at least one edge in $\overline{E}$. We also recall the following definitions (see, e.g., [70]).

Definition 1.9 (first-passage walk)**.** *Let $w \in V$. A* first-passage walk (FPW) *of length r from i to w is an r-walk of the form $i = i_0, i_1, \ldots, i_r = w$ such that $i_k \neq w$ for all $k = 0, \ldots, r - 1$; i.e., if it visits w only at the last step.*

Definition 1.10 ($\mathcal{E}$-avoiding walk)**.** *Let $\mathcal{E}$ be a set of (possibly virtual) edges, we say that a walk of the form $i_0, i_1, \ldots, i_r$ is an* $\mathcal{E}$-avoiding walk *if $(i_k, i_{k+1}) \notin \mathcal{E}$ for all $k = 0, \ldots, r - 1$, i.e., if it does not visit any of the edges in $\mathcal{E}$.*

[1] This condition is required to avoid self-loops in the walk.

Definition 1.11 (source-to-sink walk)**.** *We say that a walk is a* source-to-sink walk *if it connects a node with in-degree zero (a source) and a node with out-degree zero (a sink).*

For each graph G, we introduce the *adjacency matrix* A as the $n \times n$ matrix such that, entrywise,

$$(A)_{ij} = a_{ij} := \begin{cases} \omega_{ij} & \text{if } (i,j) \in E, \\ 0 & \text{otherwise,} \end{cases}$$

where $\omega_{ij} > 0$ is the *weight* of the edge (i, j).

Definition 1.12 (weighted graph)**.** *If $\omega_{ij} = 1$ for each $(i, j) \in E$, we say that a graph is* unweigthed*; weighted otherwise.*

Notice that A is symmetric if and only if each edge (i, j) is reciprocated and that A is upper triangular if and only if G is topologically sorted. Moreover, we say that a graph G is *sparse* if A has at most $O(n)$ nonzero entries. We recall that $(A^T\mathbf{1})_i = \text{ind}(i)$ and $(A\mathbf{1})_i = \text{outd}(i)$ as well as a classic result in graph theory that will be used extensively throughout this thesis.

Lemma 1.13. *$(A^r)_{ij}$ is equal to the number of r-walks from i to j.*

1.2. Few topics of linear algebra

In this section we recall few notions and results from linear algebra and numerical linear algebra that will be useful in the subsequent chapters.

Let $A \in \mathbb{R}^{n\times n}$ and let $\lambda_1, \lambda_2 \ldots, \lambda_n$ be its (possibly complex) eigenvalues, then the *spectral radius* of A is defined as $\rho(A) = \max_{i=1,\ldots,n}\{|\lambda_i|\}$.

Definition 1.14 (non-negative matrix)**.** *$A \in \mathbb{R}^{n\times n}$ is said to be* non-negative*, if $a_{ij} \geq 0$ for all $i, j = 1, \ldots, n$. If A in non-negative, we will write $A \geq 0$. If $B \in \mathbb{R}^{n\times n}$ is such that $A - B \geq 0$, we will write $A \geq B$.*

Definition 1.15 (reducible matrix)**.** *$A \in \mathbb{R}^{n\times n}$ is said to be* reducible *if there exist an $n \times n$ permutation matrix P such that*

$$P^T AP = \begin{pmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{pmatrix},$$

where A_{11}, A_{12}, and A_{22} are square matrices of appropriate sizes. If such permutation does not exist, A is said to be irreducible*.*

Remark 1.16. *Let $A \in \mathbb{R}^{n\times n}$ be the adjacency matrix of a strongly connected graph (or just connected, if G is undirected), then A is non-negative by definition and irreducible for [100, Theorem 1.18]. Consequently, by the Perron-Frobenius Theorem (see, e.g., [68, Theorem 8.4.4]) we have that $\rho(A)$ is a simple eigenvalue of A and that $\rho(A) > 0$. Moreover, there exists a vector $\mathbf{v}$ with positive entries, usually called Perron eigenvector of A, such that $A\mathbf{v} = \rho(A)\mathbf{v}$.*

1.2.1. Matrix functions

Let f be a scalar function, there are several different yet equivalent definitions of $f(A)$ with A a real or complex-valued $n \times n$ matrix. Here, we focus on a definition based on the Jordan canonical form and we refer to [65, Chapter 1] for an in-depth review of the topic.

We recall that any matrix $A \in \mathbb{C}^{n\times n}$ can be written as $A = ZJZ^{-1} = Z\,\mathrm{diag}(J_1, J_2, \ldots, J_p)Z^{-1}$, where Z is a nonsingular matrix and J_k is the Jordan block associated with λ_k, i.e., the matrix

$$J_k = J(\lambda_k) = \begin{bmatrix} \lambda_k & 1 & & & \\ & \lambda_k & 1 & & \\ & & \ddots & \ddots & \\ & & & \lambda_k & 1 \\ & & & & \lambda_k \end{bmatrix} \in \mathbb{C}^{m_k\times m_k},$$

with $m_1 + m_2 + \cdots + m_p = n$. Let $\lambda_1, \ldots, \lambda_s$ be distinct eigenvalues of A and let n_k be the *index* of λ_k, defined as the maximal size of the Jordan block associated with λ_k, we have the following definitions.

Definition 1.17. *We say that a function f is* defined on the spectrum of A *if $f(\lambda_k)$ and the derivates $f^{(j)}(\lambda_k)$ exist for all $j = 1, \ldots, n_k - 1$ and for all $k = 1, \ldots, s$.*

Definition 1.18. *Let f be defined on the spectrum of $A \in \mathbb{C}^{n\times n}$, and let $A = ZJZ^{-1}$ be its Jordan decomposition. Then*

$$f(A) := Zf(J)Z^{-1} = Z\,\mathrm{diag}(f(J_1), \ldots, f(J_p))Z^{-1},$$

where

$$f(J_k) := \begin{bmatrix} f(\lambda_k) & f'(\lambda_k) & \frac{f''(\lambda_k)}{2!} & \cdots & \frac{f^{(m_k-1)}(\lambda_k)}{(m_k-1)!} \\ & f(\lambda_k) & f'(\lambda_k) & \cdots & \frac{f^{(m_k-2)}(\lambda_k)}{(m_k-2)!} \\ & & \ddots & \ddots & \vdots \\ & & & f(\lambda_k) & f'(\lambda_k) \\ & & & & f(\lambda_k) \end{bmatrix} \in \mathbb{C}^{m_k \times m_k}.$$

If A is diagonalizable, then Definition 1.17 simply requires that the values $f(\lambda_k)$ exist for all $k = 1 \ldots, s$, and Definition 1.18 reduces to

$$f(A) = X f(\Lambda) X^{-1} = X \operatorname{diag}(f(\lambda_1), \ldots, f(\lambda_n)) X^{-1},$$

where $\Lambda = \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$ is a diagonal matrix having the eigenvalues of A and X is an invertible matrix whose columns are eigenvectors of A.

Now let us assume (as will often be the case in this thesis) that f admits a Maclaurin series representation, i.e., f is analytic on some open neighborhood of 0 with (finite or inifinte) radius of convergence R_f. Then the following theorem gives us necessary and sufficient conditions under which $f(A)$ can be written as a matrix power series.

Theorem 1.19. ([65, Theorem 4.7]) *Let $A \in \mathbb{C}^{n\times n}$ be a matrix with distinct eigenvalues $\lambda_1, \ldots, \lambda_s$ and let f be such that $f(z) = \sum_{p=0}^{\infty} c_p z^p$ for any $|z| < R_f$. Then $f(A)$ is well defined and*

$$f(A) = \sum_{r=0}^{\infty} c_p A^p \tag{1.1}$$

if and only if $\lambda_1, \ldots, \lambda_s$ satisfy one of the following:

1. *$|\lambda_k| < R_f$ for all $k = 1, \ldots, s$;*
2. *$|\lambda_k| = R_f$ and the series for $f^{(n_k-1)}(\lambda)$ is convergent at $\lambda = \lambda_k$, for all $k = 1, \ldots, s$.*

Notice that, if f is analytic with radius of convergence R_f, then f satisfies the hypotheses of Theorem 1.19 for all $A \in \mathbb{C}^{n\times n}$ such that $\rho(A) < R_f$. In particular, if f is entire, this holds for all $A \in \mathbb{C}^{n\times n}$ since $R_f = +\infty$.

Remark 1.20. *If f satisfies the hypotheses of Theorem 1.19 for $Y \geq 0$, then the same hypotheses hold for every X such that $0 \leq X \leq Y$. This fact is a consequence of [68, Theorem 8.1.18], which ensures that $\rho(X) \leq \rho(Y)$.*

The following well-known fact can be obtained as a corollary of Theorem 1.19 when $f(z) = (1-z)^{-1}$.

Corollary 1.21. *Let $A \in \mathbb{C}^{n\times n}$ be such that $\rho(A) < 1$, then*

$$\sum_{k=0}^{\infty} A^k = (I - A)^{-1}.$$

Computational aspects The computation of a matrix function $f(A)$ is a delicate issue, especially if A is very large or non-diagonalizable. In fact, numerically stable methods for generic functions f and generic matrices A, such as those based on the Schur decomposition (or the eigendecomposition, if A is symmetric) require a number of arithmetic operations that is at least of order of n^3. This computational cost makes those methods impractical for large-scale problems, for instance when n exceeds a few thousand. Moreover, $f(A)$ is in general a dense matrix even when A is sparse, which can bring additional issues due to memory limitations. Function-specific methods for computing $f(A)$, such as the scaling and squaring method for computing the matrix exponential also exhibit similar problems (see [65]).

In practical applications it is often not necessary to compute the whole matrix function $f(A)$. Instead, the focus is typically on evaluating (or just estimating):

- the action $f(A)\mathbf{v}$ of a matrix function over a vector $\mathbf{v} \in \mathbb{R}^n$;
- a small subset of the entries of $f(A)$; or
- quadratic forms like $\mathbf{v}^T f(A)\mathbf{v}$ for some $\mathbf{v} \in \mathbb{R}^n$.

For these tasks, we have access to a wide range of numerical methods that avoid forming $f(A)$ explicitly, which can lead to significant savings in both computational and memory costs. Examples of such methods include polynomial and rational Krylov subspaces methods (see, e.g., [25, 60, 92]), quadrature rule-based approximations of $\mathbf{v}^T f(A)\mathbf{v}$ for suitable functions of symmetric matrices (see [55]), and other function-specific methods for computing the action of a matrix function on a vector (see, e.g., [1] for the case of $e^A\mathbf{v}$).

In the next section we show that some key quantities used in in network analysis can be written in terms of entries (or sums of entries) of a suitable matrix function $f(A)$, where A is the adjacency matrix of the graph. On one hand, real-world networks are typically very large, so computing these quantities can be expensive even if A is sparse. On the other hand, high numerical

precision is seldom needed when dealing with centrality and communicability measures (see, e.g., [19]), so bounds on selected entries of $f(A)$ or $f(A)\mathbf{v}$ are often sufficient in practical applications. See also [18, 26] for an application of matrix functions for modelling non-local diffusion on graphs.

1.3. Few topics of network science

Complex networks can be modelled and studied using graphs [86]. Typical problems in network science address issues related to centrality (which nodes have most importance or influence within the network) and connectivity (how well two nodes or groups of nodes are connected to one another).

In this section we review some popular centrality and communicability measures used in the analysis of complex networks.

1.3.1. Centrality measures

Centrality measures are standard tools in network analysis. They are defined as functions

$$c : V \longrightarrow \mathbb{R}_{\geq 0}$$

that assign a non-negative score to each node in a graph to quantify its importance or influence. Given a centrality vector $\mathbf{c} = (c_i) \in \mathbb{R}^n$, we can sort its entries in non-increasing order to rank the nodes from most to least important according to the chosen metric. Many centrality measures have been introduced over the years, each tailored to capture specific properties of the nodes. However, no single centrality index fits all scenarios and the choice should be aligned with the specific application (see, e.g., the recent surveys [13, 35, 40]).

Classic examples [52] include *degree centrality*, which gives importance to a node based on the number of connections it has, i.e., its degree (Definition 1.3); *betweenness centrality*, which assigns each node a value based on the fraction of shortest paths that pass through it, highlighting nodes more likely to participate in information exchange between distant nodes; and *closeness centrality*, defined for all node i as

$$C_i = \frac{1}{\sum_{u=1}^{n} \operatorname{dist}(i, u)},$$

which favours nodes that are, on average, closer to the others.

Eigenvector centrality is another classic example of a centrality measure [28]. For a connected and undirected graph, it can be defined as the ith entry of the Perron eigenvector of the adjacency matrix A (see Remark 1.16), i.e. the vector $\mathbf{v} = (v_i)$ such that

$$A\mathbf{v} = \lambda_1 \mathbf{v},$$

where $\lambda_1 > 0$ is the largest eigenvalue of A. Observe that, since

$$v_i = \frac{1}{\lambda_1}\mathbf{e}_i^T A\mathbf{v} = \frac{1}{\lambda_1}\sum_{j\sim i} v_i,$$

we have that eigenvector centrality of node i depends on the centrality of all the nodes j adjacent to it. Thus, eigenvector centrality can be seen as a generalization of degree centrality that takes into account more than just the number of connections a node has but also the importance of its neighbouring nodes. We note in passing that the famous *Pagerank* is a variant of eigenvector centrality applied to the directed graph of the World Wide Web (see [29, 54]).

Katz centrality One important class of centrality measures is defined in terms of entries (or sums thereof) of the formal series

$$\sum_{r=0}^{\infty} c_r A^r = c_0 I + c_1 A + c_2 A^2 + \cdots \tag{1.2}$$

where $c_r > 0$ are weights. The above expression has a nice interpretation in terms of walks taking place around the network, thanks to Lemma 1.13, and formalizes a class of measures commonly referred to as *walk-based centrality measures*; see [19, Section 3] and references therein. Depending on the choice of weights c_r and on the spectral properties of A, the formal series may converge to a matrix function $f(A)$ (see Theorem 1.19). When the coefficients c_r are chosen such that $c_r = 1/r!$, we can get (1.2) to converge to the matrix exponential. In such cases, we can get the *subgraph centrality* [49], which is defined as $\exp(A)_{ii}$, or the *exponential-based total communicability* [21], which is defined as $[\exp(A)\mathbf{1}]_i$. When the c_r in (1.2) are chosen such that $c_r = \alpha^r$ for some $\alpha > 0$, the induced centrality measure is known as *Katz centrality*. Introduced in [73], it assigns to node i the ith entry of the vector

$$\mathbf{x} := (I + \alpha A + \alpha^2 A^2 + \cdots)\mathbf{1} = \sum_{r=0}^{\infty} \alpha^r A^r \mathbf{1}. \tag{1.3}$$

Notice that, thanks to Corollary 1.21, selecting $\alpha > 0$ in such a way that $0 < \alpha\rho(A) < 1$ ensures that the Katz centrality vector $\mathbf{x}$ can be written as the solution of the following linear system:

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

Remark 1.22. *If G is undirected and $0 < \alpha\rho(A) < 1$, then $(I - \alpha A)$ is a symmetric and positive definite matrix.*

Katz centrality is probably one of the most popular walk-based centrality indices in network science and has successfully been applied to a wide range of problems, such as, for example, the identification of potential disease genes in protein-interaction networks [102] or the optimization of artificial neural networks [79]. This popularity of this measure is due to its interpretability in terms of walks around a graph but also to the fact that is potentially efficient to compute [89], supported by a well-developed set of numerical linear algebra tools for solving linear systems (see, e.g., [25, 37, 93]). Moreover, Katz centrality can be easily extended to the setting of dynamic networks [58] or specialised to only consider particular types of walks [57] without affecting its easiness of computation, differently to other indices. In particular, Katz centrality is the only centrality measure that preserves the property of correctly weighting walks of any given length in temporal networks (see [11, Section 3.1]).

1.3.2. Communicability measures

The concept of communicability was first introduced in [45] as a way to quantify the connectivity of a network, which can be defined as the ability of a network to make information flows through it. The underlying idea is that a network is "well connected" if each pair of nodes i and j is joined by multiple short walks. Remarkably, given a graph with an adjacency matrix A, we can easily count all the walks that connect i and j of any length by taking powers of A. In fact, it is well known that the quantity $(A^r)_{ij}$ is equal to the number (or the weighted sum, if the graph is weighted) of all the walks of length r from i to j (see Lemma 1.13), which means that the (i, j)th entry of the power series

$$\sum_{r=0}^{\infty} c_r A^r = c_0 + c_1 A + c_2 A^2 + \cdots , \tag{1.4}$$

where $c_r \geq 0$ are chosen so that the series converges, can be considered as a weighted sum of all the walks that connect i and j in which each walk of length r is weighted by an attenuation factor that depends on r. As already mentioned, by choosing coefficients c_k equal to the Taylor expansion coefficients of a suitable analytic function f (see Theorem 1.19), we can get (1.4) to converge to the matrix function $f(A)$. More precisely, let us consider the following class of analytic functions defined via power series [22]:

$$\mathcal{P} = \Big\{ f(z) = \sum_{p=0}^{\infty} c_p z^p,\ |z| < R_f,\ c_p > 0 \text{ for all } p \geq 0 \Big\},$$

where and R_f is the (finite or infinite) radius of convergence of the series, and let A be the adjacency matrix of a graph G. We have the following definition [47].

Definition 1.23 (f-communicability)**.** *Let f be a function in $\mathcal{P}$ that satisfies the assumptions of Theorem 1.19 for A. Then we define f-*communicability *between two nodes i and j as*

$$f(A)_{ij} = \sum_{r=0}^{\infty} c_p (A^p)_{ij}. \tag{1.5}$$

Since $(A^p)_{ij}$ is equal to the number of p-walks from i to j, we can interpret $f(A)_{ij}$ as the weighted sum of all the walks that connect i and j, where the walks of length p are weighted by a penalty factor of $c_p \geq 0$. The higher the value of $f(A)_{ij}$, the greater is the number of ways in which information can travel from i to j. Therefore $f(A)_{ij}$, which is non-negative for each pair of nodes i and j since $A^p \geq 0$ and $c_p > 0$ for all $p \geq 1$, can be interpreted as a measure of how effectively i and j communicate with each other. Observe that in a directed network $f(A)_{ij} \neq f(A)_{ji}$ in general, which reflects the fact that travelling from node i to node j may be more difficult than going from j to i.

In theory, there are no further conditions that an $f \in \mathcal{P}$ must meet to induce a communicability measure on a graph G other than those given by Theorem 1.19. Nevertheless, analytic functions used in network science show in general decreasing Taylor coefficients, i.e., they can be written as $f(z) = \sum_{p=0}^{\infty} c_p z^p$ where $c_{p+1} \leq c_p$ for all $p \geq 0$. This property allows us to assign greater weights to shorter walks rather than longer ones when computing $f(A)_{ij}$ as we expect information to flow more easily through short walks than through long ones [47].

In some practical applications, it is useful to account for external interferences that may affect node communication within the network. This can be done by introducing a parameter that adjusts the weight assigned to each walk, and in such cases, we can define a *parametric f-communicability* between i and j as

$$f(\gamma A)_{ij} = \sum_{r=0}^{\infty} c_p \gamma^p (A^p)_{ij},$$

where $\gamma \in (0, \gamma^*)$ is chosen such that the above series converges. Notice that if $R_f = +\infty$ we can choose $\gamma^* = +\infty$. In physical terms, $\gamma > 0$ is usually called *inverse temperature* and can represent the amount of stress or temperature a network is exposed to [45].

By far the most popular choices of $f \in \mathcal{P}$ for defining a communicability measure on graphs are $f(z) = (1-z)^{-1}$ and $f(z) = \exp(z)$. These choices lead, respectively, to the well-studied subgraph (or exponential-based) communicability [45] and resolvent-based communicability [22]. However, a family of communicability measures which interpolates between the two aforementioned measures has recently gained interest from the network science community [8]. Such a family of measures is defined using the Mittag-Leffler functions [82], that is,

$$E_{\alpha,\beta}(z) = \sum_{p=0}^{\infty} \frac{\beta^p}{\Gamma(\alpha p + 1)} z^p, \ \alpha \in (0,1), \beta > 0,$$

where $\Gamma(z)$ is the Euler Gamma function [90, Section 5]:

$$\Gamma(z) = \int_0^{+\infty} t^{z-1} e^{-t}\, dt.$$

Notice that $E_{1,1}(z) = \exp(z)$ for all $z \in \mathbb{C}$, while $E_{0,1}(z) = (1-z)^{-1}$ for all z such that $|z| < 1$, this means that both the exponential and the resolvent-based communicability measures are induced by Mittag–Leffler functions.

Closely related to the notion of f-communicability, there are the notions of

- *f-centrality*, defined as $f(A)_{ii}$, and
- *f-total communicability*, defined as $[f(A)\mathbf{1}]_i$.

These quantities can be used to measure the importance of a node i within the network, i.e., as centrality measures (see Section 1.3.1) as they account for the weighted sum of all walks (only closed walks in the case of f-centrality) that start from it [47]. Notice that if $f(z) = (1-\alpha z)^{-1}$ with $\alpha > 0$, then f-total communicability reduces to Katz centrality [73], and if $f(z) = \exp(z)$, then f-centrality reduces to subgraph centrality [49].

The average of f-centrality and f-total communicability do have a special meaning in network analysis, as they can be considered as measures of the connectivity of a network as a whole [19]. Customarily, they take the name of *f-based Estrada index* [45],

$$\mathrm{EE}_f(A) = \frac{1}{n}\operatorname{Tr}(f(A)) = \frac{1}{n}\sum_{i=1}^{n} f(A)_{ii}, \tag{1.6}$$

and *f-based total network communicability* [21],

$$\mathrm{TC}_f(A) = \frac{1}{n}\mathbf{1}^T f(A)\mathbf{1} = \frac{1}{n}\sum_{i,j=1}^{n} f(A)_{ij}. \tag{1.7}$$

2. Updating Katz centrality in simple graphs

In Chapter 1 we introduced Katz centrality as one of the most popular and flexible walk-based centrality indices used in network science. However, repeated calculation of the measure can still be challenging in practical applications, in which simulating the sequential removal of a set of nodes/edges is often used to study issues related to network robustness and communication bottlenecks. For large networks, it may be unfeasible to recompute the centrality scores after each elimination, so it is rather useful to devise updating strategies instead.

In this chapter, which results were recently published in [7], we present two efficient algorithms for approximately updating the vector of Katz centrality after the sequential removal of a set of nodes or edges from a simple graph.

2.1. Problem setting

In the following, we assume that the removal of an edge $\{i, j\}$ from a simple graph G (Definition 1.1) coincides with setting to zero the corresponding entries of the original adjacency matrix A. Similarly, the removal of a node i is modelled as the removal of all the edges incident to it, so that the dimension of the adjacency matrix of the new graph is the same as that of the original graph. In a real-world setting, these removal operations may correspond to a road closure within a road network or an individual eliminating all their contacts on some social media platform. From a linear algebra perspective, both these type of removal correspond to rank-2 modifications of the adjacency matrix:

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

where $\mathbf{e}_i$ denotes the ith standard basis vector and $\mathbf{a}_i = A\mathbf{e}_i$ indicates the ith column of A.

In this chapter we develop efficient numerical methods to approximate

$$\mathbf{x}^{\mathcal{S}} := \sum_{r=0}^{\infty} \alpha^r A_{\mathcal{S}}^r \mathbf{1},$$

i.e., the *updated* Katz centrality vector, where $0 < \alpha\rho(A) < 1$ and $A_{\mathcal{S}}$ is adjacency matrix of the graph obtained from G by removing all the elements in $\mathcal{S}$ (either a set of nodes or a set of edges). The goal is to obtain an approximation of $\mathbf{x}^{\mathcal{S}}$ starting from the initial centrality vector $\mathbf{x}$ (to which we assume we have access), thus avoiding solving the linear system $(I - \alpha A_{\mathcal{S}})\mathbf{x}^{\mathcal{S}} = \mathbf{1}$.

Remark 2.1. *Since $0 \leq A_{\mathcal{S}} \leq A$, [68, 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 $\sum_r \alpha^r (A_{\mathcal{S}})^r = (I - \alpha A_{\mathcal{S}})^{-1}$.*

This problem has been extensively studied (whether directly of undirectly) from the perspective of low-rank updates of matrix functions and sensitivity of their entries; see, e.g., [17, 23, 91, 38, 95, 96]. For Katz centrality in particular, a few ad-hoc strategies have been proposed. In [84] the authors proposed to update centralities via an iterative refinement method applied to the original Katz vector, while [85] 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 (Definition 1.9) and of the newly introduced $\mathcal{F}$-avoiding first-passage walks (see Definition 2.9 below).

Remark 2.2. *Throughout this chapter, we will assume that all graphs are connected. This simplifying assumption does not weaken our results. 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.*

2.2. Walk counting in a simple graph

Let $\mathbf{c}_r^{\mathcal{S}} \in \mathbb{R}^n$ be the vector whose ith 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 $i = 1, \ldots, n$. Notice that, entrywise, it holds that $A^r\mathbf{1} \geq \mathbf{c}_r^{\mathcal{S}}$.

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

Lemma 2.3. *For all $r \geq 0$,*

$$\mathbf{c}_r^{\mathcal{S}} = (A^r - A_{\mathcal{S}}^r)\mathbf{1}. \tag{2.1}$$

Proof. The result follows trivially from Lemma 1.13. □

The rest of this section is devoted to finding alternative expressions for the computation of $\mathbf{c}_r^{\mathcal{S}}$ for all $r \geq 0$.

2.2.1. Counting walks through a set to edges

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

$$\mathbf{c}_r^{\{e\}} = \left(\sum_{k=0}^{r-1} (A_{\mathcal{E}})^k \left(\mathbf{e}_u \mathbf{e}_v^T + \mathbf{e}_v \mathbf{e}_u^T \right) A^{r-k-1} \right) \mathbf{1}, \tag{2.2}$$

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 $k < \ell_u := \operatorname{dist}(i, u)$ (resp., $k < \ell_v := \operatorname{dist}(i, v)$), so that the above can be rewritten entrywise as

$$(\mathbf{c}_r^{\{e\}})_i = \mathbf{e}_i^T \left(\sum_{k=\ell_u}^{r-1} (A_{\mathcal{E}})^k \mathbf{e}_u \mathbf{e}_v^T A^{r-k-1} + \sum_{k=\ell_v}^{r-1} (A_{\mathcal{E}})^k \mathbf{e}_v \mathbf{e}_u^T A^{r-k-1} \right) \mathbf{1}. \tag{2.3}$$

Secondly, if $r \leq \operatorname{dist}(i, e) := \min\{\ell_u, \ell_v\}$, then $r - 1 < \operatorname{dist}(i, e)$ and thus both summations in the right-hand side of (2.3) are zero. Therefore we can assume that $r \geq \operatorname{dist}(i, e) + 1$ whenever we are working on a specific $i \in V$.

Proof of Proposition 2.4. We proceed entrywise. Let $i \in V$ and let $\ell_x = \operatorname{dist}(i, x)$ 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 (2.2). 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. □

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 2.5. *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,*

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

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 2.4, and is therefore omitted.

Remark 2.6. *From the definition of $A_{\mathcal{E}}$ it immediately follows that*

$$\mathbf{c}_r^{\mathcal{E}} = \sum_{k=0}^{r-1} (A_{\mathcal{E}})^k A^{r-k} \mathbf{1} - \sum_{k=0}^{r-1} (A_{\mathcal{E}})^{k+1} A^{r-k-1} \mathbf{1}. \tag{2.4}$$

2.2.2. Counting walks through a set of nodes

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 2.3.1). 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.

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

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

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

$$\mathbf{c}_r^{\{w\}} = (\mathbf{e}_w^T A^r \mathbf{1})\, \mathbf{e}_w + \sum_{k=0}^{r-1} (\mathbf{e}_w^T A^{r-k-1} \mathbf{1}) (A_{\mathcal{N}})^k A \mathbf{e}_w. \tag{2.5}$$

The terms $(A_{\mathcal{N}})^k A\mathbf{e}_w$ appearing in (2.5) are of particular interest to us. For any given $k = 0, \ldots, r-1$, 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 (see Definition 1.9). 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 2.4.

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

$$(\mathbf{q}_k)_i = \begin{cases} 0 & \text{if } 0 \leq k < \operatorname{dist}(i,w), \\ (A_{\mathcal{N}}^{k-1} A \mathbf{e}_w)_i & \text{if } k \geq \operatorname{dist}(i,w). \end{cases} \tag{2.6}$$

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, (2.5) 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 2.7. *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*

$$\mathbf{c}_r^{\{w\}} = \sum_{k=0}^{r} (\mathbf{e}_w^T A^{r-k} \mathbf{1}) \mathbf{q}_k. \tag{2.7}$$

Let $\ell_i = \operatorname{dist}(i,w)$, then (2.7) can be written entrywise for all $i \in V$ as

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

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, 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 2.2.2 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. This mismatch

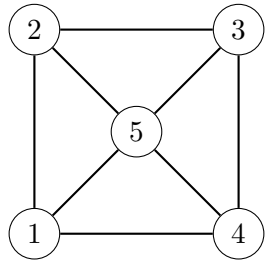


Figure 2.1: Toy network.

arises because the nodes that we want our walks to visit are connected to each other, causing walk $1 \to 2$ to be counted twice in the right hand side of the above expression. Thus, a different approach should be devised that correctly counts walks when adjacent 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 $\mathcal{N} = \{w_1, w_2, \ldots, w_s\}$ 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 $\mathcal{N}$;
- $\mathcal{E}^{\text{out}} = \{\{w, j\} \in \mathcal{E} : w \in \mathcal{N} \text{ and } j \in V \backslash \mathcal{N}\}$ is the subset of $\mathcal{E}$ containing edges connecting a node in $\mathcal{N}$ with a node in $V \backslash \mathcal{N}$, i.e., it is $\mathcal{E}^{\text{out}} = \mathcal{E} \backslash \mathcal{E}^{\text{in}}$;
- 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 2.2.2 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 2.2. We point out that $A = A_{\mathcal{N}} + A^{\text{in}} + A^{\text{out}}$.

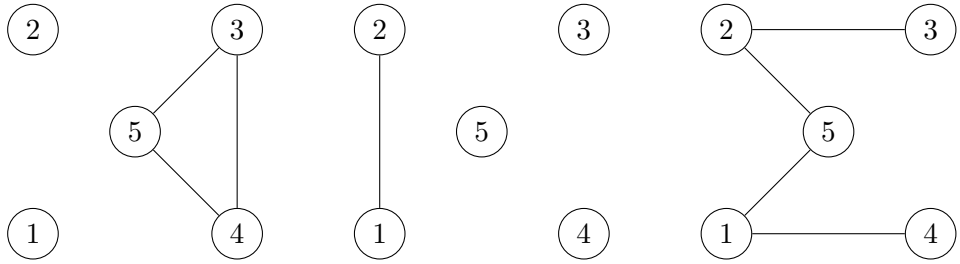


Figure 2.2: Graphs corresponding to the adjacency matrix of $A_{\mathcal{N}}$ (left), A^{in} (middle) and A^{out} (right) when $\mathcal{N} = \{1, 2\}$ from the network in Figure 2.2.2.

We are now in a position to derive an explicit formula for $\mathbf{c}_r^{\mathcal{N}}$ for $r > 0$. By Proposition 2.5 it follows

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

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

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

we derive

$$\mathbf{c}_r^{\mathcal{N}} = A^{\text{in}}A^{r-1}\mathbf{1} + \sum_{k=0}^{r-1} (A_{\mathcal{N}})^k A^{\text{out}} A^{r-k-1}\mathbf{1}. \tag{2.8}$$

Equation (2.8) 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 adjacent nodes. The second term, on the other hand, includes walks that avoid nodes in $\mathcal{N}$ for the first k steps, than 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 adjacent nodes in $\mathcal{N}$ are consecutively visited, as long as at least one step is taken before visiting the edge connecting them.

Remark 2.8. *When there are no adjacent nodes in $\mathcal{N}$, i.e., when $\mathcal{E}^{in} = \emptyset$, it holds that $A^{out} = A - A_{\mathcal{N}}$ and* (2.8) *reduces to* (2.4).

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 2.10 below).

Definition 2.9. *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 $i = i_0, i_1, \ldots, i_{r-1}, i_r = w$ such that $i_k \notin \mathcal{F}$ for all $k = 0, \ldots, r-1$.*

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 adjacent 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 ith 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 $0 < k < \mathrm{dist}(i, w)$ it immediately follows that $(\mathbf{q}_k^{\mathcal{F}})_i = 0$.

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

$$(\mathbf{q}_k^{\mathcal{F}})_i = \begin{cases} 0 & \text{if } 0 \leq k < \mathrm{dist}(i, w), \\ (A_{\mathcal{N}}^{k-1} A_{\mathcal{F}} \mathbf{e}_w)_i & \text{if } k \geq \mathrm{dist}(i, w), \end{cases} \tag{2.9}$$

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 2.10. *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*

$$(\mathbf{c}_r^{\mathcal{N}})_i = \begin{cases} \sum_{w\in\mathcal{N}} \sum_{k=1}^{r} (\mathbf{q}_k^{\mathcal{F}_w})_i [A^{r-k}\mathbf{1}]_w & \text{if } i \notin \mathcal{N} \\ (A^r\mathbf{1})_i & \text{otherwise,} \end{cases} \tag{2.10}$$

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 $k > 0$, (2.8) 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. □

Remark 2.11. *When $\mathcal{F} = \emptyset$, Definition 2.9 becomes Definition 1.9, the relationships in* (2.9) *boil down to those in* (2.6) *with $\mathcal{N} = \{w\}$, and Proposition 2.10 reduces to Proposition 2.7.*

One of the main reasons to rewrite the count in (2.8) to become (2.10) 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 2.12. *Let $\mathcal{N} = \mathcal{F} \cup \{w\}$, where $w \in V$ and $\mathcal{F} \subset V \setminus \{w\}$. Then, in the above notation,*

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

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 (2.9), for $k = 1, 2, \ldots$, we get

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

The fact that $(\mathbf{q}_k^{\mathcal{F}})_j = 0$ for all $j \in \mathcal{N}$ yields the conclusion for $k = 1, 2, \ldots$. 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 ad w, and avoiding nodes in $\mathcal{F}$, i.e., $\mathbf{q}_1^{\mathcal{F}}$. This concludes the proof. □

Corollary 2.13. *When $\mathcal{F} = \emptyset$, for all $k = 0, 1, 2, \ldots$ 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 2.12 (resp., Corollary 2.13) implies that it is possible to compute $\mathbf{q}_{k+1}^{\mathcal{F}}$ (resp., $\mathbf{q}_{k+1}$) for $k = 0, 1, 2, \ldots, r$, with r matrix vector products involving the adjacency matrix A.

2.3. Main results

In this section we show that the results obtained so far can be used to fully characterize the variation in Katz centrality for nodes (and induced total network communicability) in networks when edges (resp., nodes) are removed from the graph. In particular, we 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 [27, Proposition 5.1.b] to the case in which $\mathcal{S} = \mathcal{E} \subset E$ or $\mathcal{S} = \mathcal{N} \subset V$. This in turn will allow us to provide bounds on the decay of total communicability and thus develop numerical strategies for their update.

2.3.1. Updating Katz centrality via walk count

Let $\mathbf{x}^{\mathcal{S}} = (x_i^{\mathcal{S}})$ be the vector of Katz centralities computed on the graph G after removal of the elements in $\mathcal{S}$.

Theorem 2.14. *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*

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

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 form 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 (2.1) yields the result. □

We observe that, for any given node $i \in V$, the expressions in Theorem 2.14 can be rewritten as follows

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

where $\ell_{\mathcal{S}} = \min_{s \in \mathcal{S}}\{\mathrm{dist}(i, s)\}$, with $\mathrm{dist}(i, e) = \min\{\mathrm{dist}(i, u), \mathrm{dist}(i, v)\}$ 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 Section 2.2.1 and 2.2.2 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.

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

Corollary 2.15. *The change in the centrality score of nodes after the removal of a set of edges $\mathcal{E}$ is*

$$\mathbf{x} - \mathbf{x}^{\mathcal{E}} = \sum_{\{u,v\}\in\mathcal{E}} \left[\left(\sum_{k=0}^{\infty} \alpha^{k+1} (A_{\mathcal{E}})^k \mathbf{e}_v \right) x_u + \left(\sum_{k=0}^{\infty} \alpha^{k+1} (A_{\mathcal{E}})^k \mathbf{e}_u \right) x_v \right], \quad (2.11)$$

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 (2.4) into Theorem 2.14, 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. □

Corollary 2.16. *When $\mathcal{E} = \{e\} \subset E$ with $e = \{u, v\}$, for all $i \in V$ it holds*

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

where $\ell_w = \operatorname{dist}(i, w)$ 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 2.16 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 2.17. *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 2.16, 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

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

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. □

We consider now the situation where a set of nodes $\mathcal{N}$ is removed from the graph. Similarly to Section 2.2.2, 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 2.18. *Let $\mathcal{N} \subset V$ be a set of nodes in G, then*

$$x_i - x_i^{\mathcal{N}} = \begin{cases} \sum_{w \in \mathcal{N}} \left(x_w \sum_{r=\ell_w}^{\infty} \alpha^r (\mathbf{q}_r^{\mathcal{F}_w})_i \right), & \text{if } i \notin \mathcal{N} \\ x_i - 1 & \text{otherwise} \end{cases} \tag{2.12}$$

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 2.14 and Proposition 2.10, 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 chapter, that $\mathbf{c}_0^{\mathcal{N}} = \mathbf{0}$. □

[2]Recall that we assumed 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.

Corollary 2.19. *In the hypothesis of Corollary 2.18, if* $\mathcal{N} = \{w\}$ *then*

$$x_i - x_i^{\{w\}} = \begin{cases} x_w \sum_{r=\ell}^{\infty} \alpha^r (\mathbf{q}_r)_i, & \text{if } i \neq w \\ x_i - 1 & \text{otherwise,} \end{cases} \tag{2.13}$$

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

Corollary 2.19 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 2.20. *If* $i \sim w$ *then*

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

Proof. Using (2.13) and the fact that $\mathrm{dist}(i, w) = 1$ 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. □

2.3.2. 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 simple graph.

In this section we use the results from Corollary 2.16 and 2.19 to find an upper bound on the change of the *total network communicability* [21], here defined as

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

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

Theorem 2.21. *Given* $\mathcal{N} = \{w\} \subset V$ *it holds that*

$$TC - TC^{\{w\}} \leq \frac{1}{n}(x_w^2(1 - \alpha^2 \mathrm{deg}(w)) - 1), \tag{2.14}$$

where $\mathrm{deg}(w)$ *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 2.15 and symmetry of $A_{\mathcal{E}}$, it follows that

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

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

$$x_w = 1 + \alpha\sum_{v:\, v\sim w}x_v, \tag{2.15}$$

we get

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

The fact that $\mathbf{x}^{\mathcal{E}} = \mathbf{x}^{\{w\}}$, Proposition 2.20, and (2.15) imply

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

from which (2.14) immediately follows. □

Theorem 2.22. *Let $e = \{u, v\}$ and consider $\mathcal{E} = \{e\}$. Then*

$$TC - TC^{\{e\}} \le \frac{1}{n}\left(2\alpha x_u x_v - \alpha^2 x_u^2 - \alpha^2 x_v^2\right). \tag{2.16}$$

Proof. From Corollary 2.16 and symmetry of $A_{\mathcal{E}}$ we obtain

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

Now by applying Proposition 2.17 we get

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

from which the conclusion straightforwardly follows. □

Remark 2.23. *Consider now the function* $f_\alpha(x,y) = 2\alpha xy - \alpha^2x^2 - \alpha^2y^2$ *appearing in* (2.16) *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 2.22 can be used to devise strategies for optimizing total network communicability through edge removal, similar to the ones in [6]. 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 [6] for solving the* downdate problem *considered there.*

2.4. Algorithms

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 $\widehat{\mathbf{x}}$.

2.4.1. 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

Algorithm 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, $\alpha \in (0, 1/\rho(A))$, $L^{\mathcal{N}}_{\max} \in \mathbb{N}$, $\texttt{tol} \in \mathbb{R}$, 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.

$\widehat{\mathbf{x}}^{\{w\}} = \mathbf{x}$;
$L = 1$;
$\mathbf{q} = \alpha * A * \mathbf{e}_w$;
$\widehat{\mathbf{x}}^{\{w\}} = \widehat{\mathbf{x}}^{\{w\}} - x_w * \mathbf{q}$;
while $x_w\|\mathbf{q}\|_2 \,/\, \|\mathbf{x}\|_2 > \texttt{tol}$ **and** $L < L^{\mathcal{N}}_{\max}$ **do**
  $\mathbf{q} = \alpha * A * \mathbf{q}$;
  $q_w = 0$;
  $\widehat{\mathbf{x}}^{\{w\}} = \widehat{\mathbf{x}}^{\{w\}} - x_w * \mathbf{q}$;
  $L = L + 1$;
end while
$(\widehat{\mathbf{x}}^{\{w\}})_w = 1$

in Corollary 2.19 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

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

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 $\widehat{x}_w^{\{w\}} = x_w^{\{w\}} = 1$. A pseudocode for our proposed algorithm can be found in Algorithm 1.

Given that $\mathbf{x}$, the vector of Katz centralities, is known, the computational effort to generate the approximant $\widehat{\mathbf{x}}^{\{w\}}$ is spent computing the vectors $\mathbf{q}_r$ for $r = 1, \ldots, L$. Thanks to the result presented in Corollary 2.13, 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 walk-length that we want to allow, denoted by $L^{\mathcal{N}}_{\max}$, the parameter L is chosen as the smallest $0 < L \leq L^{\mathcal{N}}_{\max}$ such that

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

Algorithm 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, $\alpha \in (0, 1/\rho(A))$, $L^{\mathcal{E}}_{\max} \in \mathbb{N}$, $\texttt{tol} \in \mathbb{R}$, $e = \{u, v\}$ edge to be removed.

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

$\widehat{\mathbf{x}}^{\{e\}} = \mathbf{x} - \alpha * x_v * \mathbf{e}_u - \alpha * x_u * \mathbf{e}_v$;
$L = 1$;
$\mathbf{s} = \alpha^2 * (A * \mathbf{e}_u - \mathbf{e}_v)$;
$\mathbf{t} = \alpha^2 * (A * \mathbf{e}_v - \mathbf{e}_u)$;
$\widehat{\mathbf{x}}^{\{e\}} = \widehat{\mathbf{x}}^{\{e\}} - x_v * \mathbf{s} - x_u * \mathbf{t}$;
while $\|x_v * \mathbf{s} + x_u * \mathbf{t}\| \,/\, \|\mathbf{x}\|_2 > \texttt{tol}$ **and** $L < L^{\mathcal{E}}_{\max}$ **do**
  $\mathbf{s} = \alpha * (A * \mathbf{s} - s_u * \mathbf{e}_v - s_v * \mathbf{e}_u)$;
  $\mathbf{t} = \alpha * (A * \mathbf{t} - t_u * \mathbf{e}_v - t_v * \mathbf{e}_u)$;
  $\widehat{\mathbf{x}}^{\{e\}} = \widehat{\mathbf{x}}^{\{e\}} - x_v * \mathbf{s} - x_u * \mathbf{t}$;
  $L = L + 1$;
end while

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 $\texttt{tol}$.

2.4.2. 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 2.16 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 $L^{\mathcal{E}}_{\max}$ and then select $0 < L \leq L^{\mathcal{E}}_{max}$ dynamically as the smallest index such that

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

Algorithm 2 contains the pseudocode for the proposed procedure.

2.4.3. Computational cost

The following holds.

Proposition 2.24. *For a sparse network, the cost of Algorithm 1 and 2 is $\mathcal{O}(Lm)$ operations, where m is the number of undirected edges in the original graph.*

Proof. For Algorithm 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 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 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. □

Clearly, Algorithm 1 and 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 (2.11) (resp., (2.12)) to L terms. In this case it is easy to see that, under the same assumption of Proposition 2.24, the generalized algorithms would have a computational complexity in time of $O(|\mathcal{S}|Lm)$, where $\mathcal{S}$ is the set of elements being removed. In addition, since (2.11) and (2.12) 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 $O(Lm)$ per processor.

2.5. Numerical experiments

In this section, we compare the performance of Algorithm 1 and 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 $\alpha = 0.85/\rho(A)$ as in [21].

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 [64]. 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 2.1. Recall that the diameter of a network is $\text{diam} = \max_{ij} \text{dist}(i, j)$,

Table 2.1: 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 2.2: 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

while the average eccentricity is $\epsilon = \frac{1}{n}\sum_i \max_{j\in V}\{\text{dist}(i,j)\}$. We only considered connected networks for simplicity.

The real-world networks in our dataset, both available at [36], 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 2.2.

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.

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 $A_{\mathcal{S}}$ of the new graph, we compute/approximate the updated Katz centrality $\mathbf{x}^{\mathcal{S}}$ in three different ways;

(i) `pcg` ($\mathbf{x}^{(0)} = \mathbf{v}$): solve $(I-\alpha A_{\mathcal{S}})\mathbf{y} = \mathbf{1}$ 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 [51]: truncate the series $\sum_{r=0}^{\infty}(A_{\mathcal{S}})^r\mathbf{1}$ with a tolerance of 10^{-4} on the stopping criteria and a maximum number of iteration of 100 (see [51] for details); and

(iii) Algorithm 1 or Algorithm 2 with $L_{\max}^{\mathcal{S}} = 30$ and $\texttt{tol} = 10^{-4}$.

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 $\texttt{tol} = 10^{-4}$. 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 2.25. *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.

In view of Proposition 2.25, in our experiments we used a relative tolerance of $\texttt{tol}_{\texttt{pcg}} := \texttt{tol}/10 = 10^{-5}$ for the `pcg` algorithm in method (i), where $\texttt{tol} = 10^{-4}$ 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 2.1.

Lastly, we want to mention in passing that we do not compare our proposed strategy with the one described in [84] since experiments not shown here returned results comparable with those obtained by `pcg` ($\mathbf{x}^{(0)} = \mathbf{0}$).

In Table 2.3 and 2.5, we report the average number of iterations and average CPU times needed to compute an approximation to $\mathbf{x}^{\mathcal{S}}$ using the methods described. All values are averaged over 30 runs with the same parameters.

Table 2.3: Number of iterations and CPU timings required for approximating $\mathbf{x}^{\mathcal{S}}$ with methods (i)–(iii) after the removal of one node selected uniformly at random. The chosen tolerances are `tol`$= 10^{-5}$ for method (i), and `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	`pcg`$(\mathbf{x}^{(0)} = \mathbf{0})$		`pcg`$(\mathbf{x}^{(0)} = \mathbf{x})$		method in [51]		Algorithm 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**	$2.42{\cdot}10^{-3}$

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 `pcg`$(\mathbf{x}^{(0)} = \mathbf{x})$. This behaviour can be explained by observing that, while `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 `pcg`$(\mathbf{x}^{(0)} = \mathbf{x})$ are initialized with $\mathbf{x}^{(0)} = \mathbf{x}$, which may already be a good approximation to $\mathbf{x}^{\mathcal{S}}$. In fact, thanks to Corollary 2.19 and 2.16, we know that if the removed node (resp., edge) has (resp., joins nodes with) a relative small centrality score(s), then $\mathbf{x}^{\mathcal{S}}$ 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 $\mathbf{x}^{\mathcal{S}}$ by counting from scratch all the walks that *avoid* s; 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 Algorithm 1 and 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

Table 2.4: Number of iterations and CPU timings required for approximating $\mathbf{x}^{\mathcal{S}}$ with methods (i)–(iii) after the removal of one edge selected uniformly at random. The chosen tolerances are `tol`$= 10^{-5}$ for method (i), and `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	`pcg`($\mathbf{x}^{(0)} = \mathbf{0}$)		`pcg`($\mathbf{x}^{(0)} = \mathbf{x}$)		method in [51]		Algorithm 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**	$1.16{\cdot}10^{-3}$

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, 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 Algorithm 1 and 2. We also look at the *intersection similarity* of the induced rankings [50]. The intersection similarity of two ranking vectors $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ is defined as

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

where, using MATLAB notation, we denote by $\boldsymbol{\beta}_{1:i}$ and $\boldsymbol{\gamma}_{1:i}$ the vectors of length i containing the first i elements in $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$, 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., [21, 33]). If $isim(\boldsymbol{\beta}, \boldsymbol{\gamma}) \approx 0$, then the two rankings are almost identical, whereas $isim(\boldsymbol{\beta}, \boldsymbol{\gamma}) = 1$ represents two completely different rankings.

Figure 2.3 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 Algorithm 2 and 1 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., [91]). We plot the results of our experiment in Figure 2.4. 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 found in Theorem 2.21 and 2.22 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 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 Figure 2.5, 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 (2.14) and (2.16). 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 the error by selectively recomputing the measures after a suitable number of iterations (see, e.g., [27] for some strategies).

2.6. Conclusions

In this chapter we derived explicit formulas for the change in Katz centrality when nodes and/or edges are removed from a simple graph. 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)

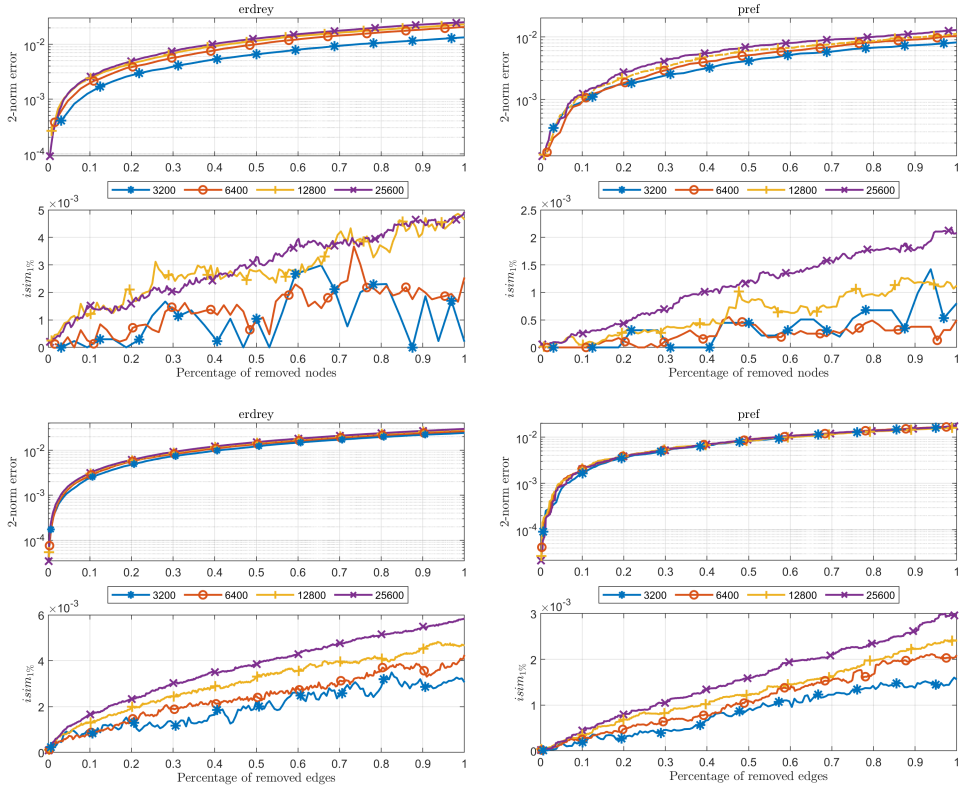


Figure 2.3: Random removal of a sequence of nodes (top figure) or edges (bottom figure). Evolution of the relative 2-norm error and $isim_p$, where $p = \lceil m/100 \rceil$, when sequentially removing edges from synthetic networks of increasing sizes.

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).

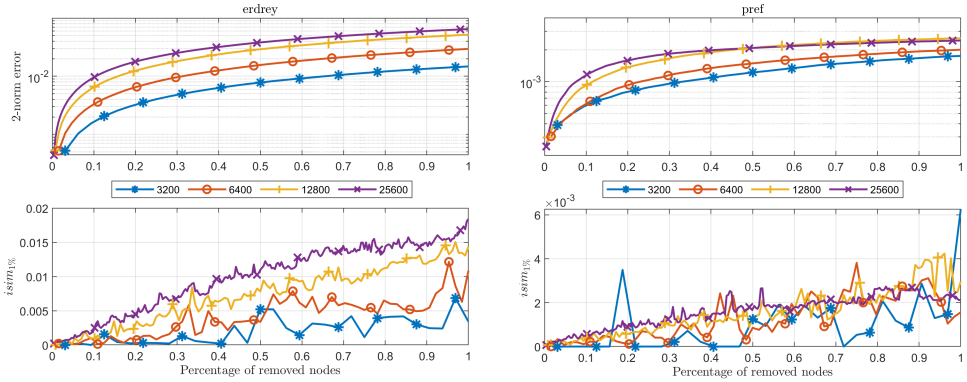


Figure 2.4: Targeted removal of a sequence of nodes. Evolution of the relative 2-norm error and $isim_p$, where $p = \lceil n/100 \rceil$, when sequentially removing 1% of the most important nodes (according to Katz centrality) from synthetic networks of increasing sizes. All the values are averaged over 30 runs.

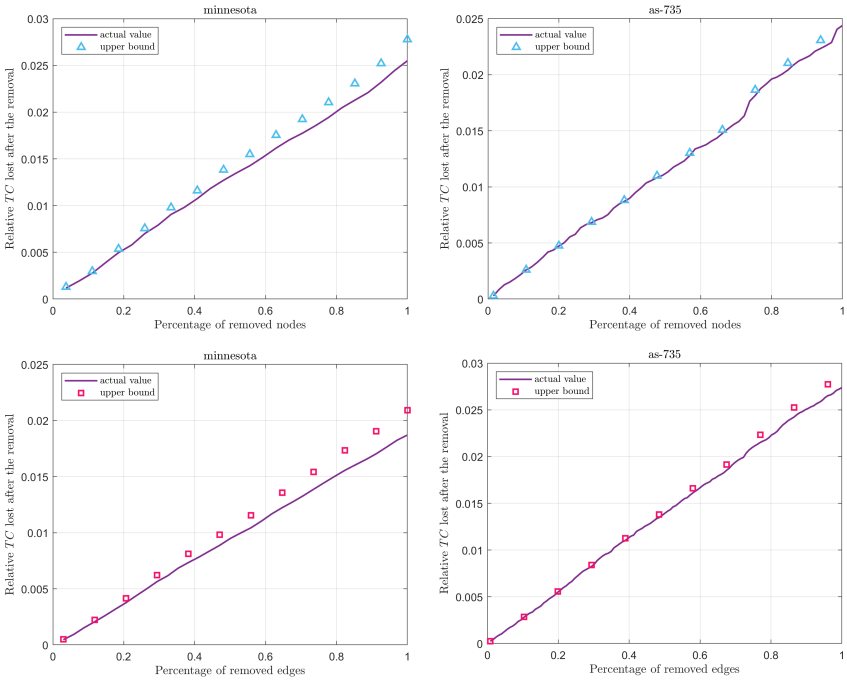


Figure 2.5: Comparison between the bound (2.14)–(2.16) and the actual value of TC lost (relative to the initial value) after the removal of 1% of the total nodes (top figure) or edges (bottom figure) form the `minnesota` and the `as-735` networks, taken uniformly at random. The values are averaged over 30 runs.

3. Estimating f-communicability after edge perturbation

In Chapter 1 we defined the f-communicability between two nodes i and j as $f(A)_{ij}$, where f is a suitable analytic function and A is the adjacency matrix of the graph. In this chapter we present bounds on the modifications of f-communicability and related network indices when a set of edges is added/removed from a graph and f belongs to a special class of analytic functions. We also address the more general case in which a perturbation affects a subset of edge weights within a weighted graph.

The material for this chapter is part of a work in preparation [9].

3.1. Problem setting and notation

Let $G = (V, E)$ be a graph, we define an *edge perturbation* in G as any modification that alters at least one of its edges or virtual edges[1]. In the first part of this chapter, we focus on two types of perturbations in an unweighted graph and we leave to Section 3.5 the case of weight perturbation in a weighted graph. For simplicity, we will assume that all the graphs in this chapter have no self-loops (edges connecting a node to itself).

Let us begin by considering the case in which a subset of edges $\mathcal{E} \subset E$ is removed from G, thus producing the new graph $G' = (V, E')$, where $E = E' \cup \mathcal{E}$. This scenario represents the situation where a random failure or an intentional disruption causes some of the connections in the graph to break. This problem, which we refer to as the *downdate problem*, has several real-life applications, such as assessing the resilience of a network by simulating failures or optimizing a network by removing unnecessary elements, thereby making it cheaper to operate [5, 6].

[1] We recall that an edge (i, j) is said to be virtual if $i \neq j$ and $(i, j) \in \overline{E}$, where $\overline{E}$ denotes the complement of $E \subset V \times V$.

The *update problem*, on the other hand, arises when a subset of virtual edges $\mathcal{E} \subset \overline{E}$ is added to G, thus obtaining a new graph $G'' = (V, E'')$, where $E'' = E \cup \mathcal{E}$. For instance, in a transportation network, this may represent the opening of a new route, or in a collaboration network, it may correspond to the establishment of a new partnership. From an optimization perspective, an organization managing a network may face this problem with the aim of identifying the "best" edges to introduce within a limited "perturbation budget" (see, e.g., [80]).

Ultimately, the update and downdate problems are two sides of the same coin; in both settings we have two graphs that are almost identical, apart from the presence in one of them of a few additional edges. Within the update problem, we model the scenario of going from the graph with fewer edges to the one with more; the downdate problem models the opposite scenario by going in the opposite direction. Indeed, the removal (resp., addition) of a set of edges $\mathcal{E}$ is equivalent to setting to zero (resp., one) the corresponding entries of the original (binary) adjacency matrix. This means that two problems discussed above can be simultaneously formalized by introducing some clever notation. To that end, we will denote by

- M the adjacency matrix of the graph with fewer edges;
- $M_{\mathcal{E}}$ the adjacency matrix of the graph with more edges.

As a clarifying example, let us consider the downdate problem in Figure 3.1, in which edge $(2, 4)$ is removed from the graph on the left, transforming it into the graph on the right. Then, in the above notation, we have that the respective adjacency matrices are

$$M_{\mathcal{E}} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & \mathbf{1} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \xrightarrow{\text{downdate}} \quad M = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & \mathbf{0} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Similarly, if we consider the update problem in Figure 3.1, in which the graph on the right is transformed into the graph on the left via the addition of the edge $(3, 4)$, we have that the corresponding adjacency matrices are

$$M = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & \mathbf{0} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \xrightarrow{\text{update}} \quad M_{\mathcal{E}} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & \mathbf{1} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

i.e., the same as before.

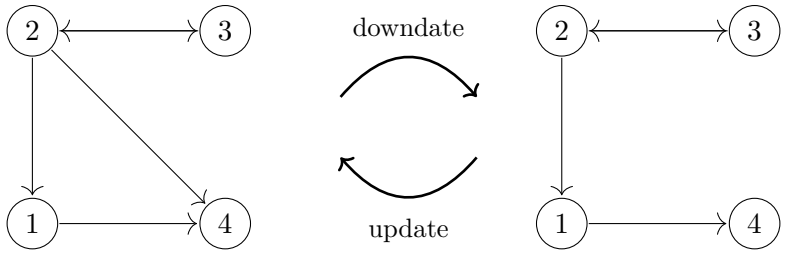


Figure 3.1: Update and downdate problem on a toy graph.

Therefore, if A is the adjacency matrix of G and $\mathcal{E} \subset E$ is a set of edges that is removed from G in the downdate problem, then

$$M_{\mathcal{E}} = A \text{ and } M = A - \sum_{e \in \mathcal{E}} \mathbf{e}_{u_e} \mathbf{e}_{v_e}^T,$$

where $e = (u_e, v_e) \in \mathcal{E}$. Similarly, if $\mathcal{E} \subset \overline{E}$ is a set of virtual edges that is added to G in the update problem, we have that

$$M = A \text{ and } M_{\mathcal{E}} = A + \sum_{e \in \mathcal{E}} \mathbf{e}_{u_e} \mathbf{e}_{v_e}^T.$$

Remark 3.1. *By construction $0 \leq M \leq M_{\mathcal{E}}$, regardless of whether $\mathcal{E} \subset E$ or $\mathcal{E} \subset \overline{E}$.*

In this chapter we estimate the changes in the values of f-communicability (Definition 1.23) after the removal (resp., addition) of a set of edges $\mathcal{E}$ from (resp., to) a graph. In particular, we develop explicit bounds on the entries of

$$f(M_{\mathcal{E}}) - f(M), \tag{3.1}$$

where $M_{\mathcal{E}}$ and M are the adjacency matrices defined above, and f belongs to a special class of analytic functions. These bounds will allow us to quantify the decrease in communicability caused by the removal of a set of edges in the downdate problem, or the improvements resulting from the addition of a set of virtual edges in the update problem. The goal is to avoid computing the full matrix $f(M)$ or $f(M_{\mathcal{E}})$ for each possible set of "perturbed edges" $\mathcal{E}$, which would be unfeasible when n is large (see Section 1.2.1).

Our approach to estimate (3.1) will build upon a result that enumerates all (directed) walks that visit at least one of the elements in $\mathcal{E}$ and will rely on the assumption that we have access to the values of f-communicability in the graph before the edge perturbation, which means having full knowledge of $f(M)$ or $f(M_{\mathcal{E}})$ depending on whether $\mathcal{E} \subset E$ or $\mathcal{E} \subset \overline{E}$.

Table 3.1: A scheme summarising the notation introduced in Section 3.1. Here, the symbols ✓and **?** indicate whether a given quantity is assumed to be known in a particular problem or is, instead, the objective of our analysis.

	M	$M_{\mathcal{E}}$	$f(M)$	$f(M_{\mathcal{E}})$
downdate problem	$A - \sum_{e\in\mathcal{E}} \mathbf{e}_{u_e}\mathbf{e}_{v_e}^T$	A (✓)	**?**	✓
update problem	A (✓)	$A + \sum_{e\in\mathcal{E}} \mathbf{e}_{u_e}\mathbf{e}_{v_e}^T$	✓	**?**

Table 3.1 provides an overview of the problems discussed in this section and summarises the new notation, specifying for each problem which quantities are assumed to be known. As we discuss later, we will need some additional hypotheses on the analytic function f that we are allowed to consider. Nonetheless, we will prove that the results hold for a broad class of anaytic functions that includes the family of Mittag-Leffler functions [82] (see Section 3.2 below).

3.2. A special class of communicability measures

Let us consider the following class of analytic functions:

$$\mathcal{P}_\gamma = \{f \in \mathcal{P} : f(z) = \sum_{p=0}^{\infty} c_p z^p, \\ \text{with } \gamma^{p+q} c_p c_q \le c_{p+q} \le c_p c_q \text{ for all } p, q \ge 0\},$$

where $\gamma \in [0, 1]$ and $\mathcal{P}$ is the class of analytic functions with positive Taylor coefficients defined in Section 1.3.2.

Remark 3.2. *By definition, $\mathcal{P}_1 \subseteq \mathcal{P}_{\gamma'} \subset \mathcal{P}_\gamma \subseteq \mathcal{P}_0 \subseteq \mathcal{P}$ for all $0 \le \gamma < \gamma' \le 1$.*

Given a function $f \in \mathcal{P}_\gamma$ that satisfies the assumptions of Theorem 1.19 for A, we have that f induces a communicability measure on a graph with adjacency matrix A (see Definition 1.23). The following lemma ensures that it is sufficient to check whether $c_1 \le 1$ to conclude that $f \in \mathcal{P}_\gamma$ has monotonically decreasing Taylor coefficients.

Lemma 3.3. *Let f be a function in $\mathcal{P}_\gamma$ for some $\gamma \in [0, 1]$. If $c_1 \le 1$, then $c_{p+1} \le c_p$ for all $p \ge 1$.*

Proof. Since $f \in \mathcal{P}_\gamma$, it holds that $c_{p+1} \le c_1 c_p$ for all $p \ge 1$. The fact that $c_1 \le 1$ yields the conclusion. □

The next lemma is the fundamental property that will allow us to estimate the modification of f-communicability in a graph after edge perturbation in Sections 3.4 and 3.5.

Lemma 3.4. *Let $f \in \mathcal{P}_\gamma$ for some $\gamma \in [0,1]$. Then*

$$\gamma^{2p+q+2} c_1 c_p c_q \leq c_{p+q+1} \leq c_1 c_p c_q.$$

Proof. Since $f \in \mathcal{P}_\gamma$, we have that $\gamma^{p+q+1} c_{p+1} c_q \leq c_{p+q+1} \leq c_{p+1} c_q$ and $\gamma^{p+1} c_1 c_p \leq c_{p+1} \leq c_1 c_p$, which together yield the conclusion. □

In the rest of this section, we present some properties of the newly introduced class of analytic functions and we show examples of functions that belong to $\mathcal{P}_\gamma$ for some $\gamma \in [0,1]$.

Proposition 3.5. *Let $\sigma > 0$ and let f be a function in $\mathcal{P}_\gamma$ for some $\gamma \in [0,1]$. Then $h_\sigma(z) := f(\sigma z) \in \mathcal{P}_\gamma$ for all $|z| < R_f/\sigma$.*

Proof. Since $\sigma > 0$, it is easy to see that $h_\sigma(z) = f(\sigma z) = \sum_{p=0}^{\infty} \sigma^p c_p z^p$ belongs to $\mathcal{P}$ and has radius of convergence $R_{h_\sigma} = R_f/\sigma$. The fact that $\sigma^{p+q} = \sigma^p \sigma^q$ and the fact that $f \in \mathcal{P}_\gamma$ yield the conclusion. □

Proposition 3.6. *Let $\alpha \in (0,1]$, then*

$$E_{\alpha,\beta}(z) = \sum_{p=0}^{\infty} \frac{\beta^p}{\Gamma(\alpha p + 1)} z^p \ \in \mathcal{P}_{1/2^\alpha} \quad \text{for all } \beta > 0,$$

where $\Gamma(z)$ is the Euler Gamma function [90, Section 5].

Proof. Since $E_{\alpha,\beta}(z) = E_{\alpha,1}(\beta z)$ for all $\beta > 0$, by Proposition 3.5 it is sufficient to prove that $E_{\alpha,1} \in \mathcal{P}_{1/2^\alpha}$. The latter holds if and only if

$$\frac{1}{2^{\alpha(p+q)}} \frac{1}{\Gamma(\alpha p + 1)} \frac{1}{\Gamma(\alpha q + 1)} \leq \frac{1}{\Gamma(\alpha p + \alpha q + 1)} \leq \frac{1}{\Gamma(\alpha p + 1)} \frac{1}{\Gamma(\alpha q + 1)},$$

for all $p, q \geq 0$ or, equivalently, when

$$1 \leq \frac{\Gamma(\alpha p + \alpha q + 1)}{\Gamma(\alpha p + 1)\Gamma(\alpha q + 1)} \leq 2^{\alpha(p+q)}, \text{ for all } p, q \geq 0.$$

To prove the above inequality, we will prove a slightly stronger result. Specifically, we will prove that for each $y > 0$ and $0 \leq x \leq y$,

$$1 \leq \frac{\Gamma(y+1)}{\Gamma(x+1)\Gamma(y-x+1)} = \binom{y}{x} \leq 2^y, \tag{3.2}$$

where $\binom{y}{x}$ denotes the *continuos binomial coefficient*; see, e.g., [94]. We start by observing that $\binom{y}{x}$ has a maximum in $x = y/2$, and two minima in $x = 0$ and $x = y$ [94]. Then, the fact that $\binom{y}{y} = \binom{y}{0} = 1$ since $\Gamma(1) = 1$ yields the lower bound in (3.2). The upper bound follows from [94, Theorem 3], which gives us

$$\binom{y}{\frac{y}{2}} = \frac{2^y}{\pi} \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \cos^y t \, dt,$$

and the fact that $|\cos^y t| \leq 1$ for all $y > 0$ and $t \in [-\frac{\pi}{2}, \frac{\pi}{2}]$. □

Lemma 3.7. *Let $f \in \mathcal{P}_\gamma$ for some $\gamma \in (0, 1]$. Then $c_0 = 1$.*

Proof. Since $f \in \mathcal{P}_\gamma$, we have that $\gamma^p c_p c_0 < c_p \leq c_p c_0$ for all $p \geq 0$. Consequently, it must hold that $c_0 = (c_0)^2 = 1$. □

Proposition 3.8. *$f \in \mathcal{P}_1$ if and only if $f(z) = (1 - \sigma z)^{-1}$ for some $\sigma > 0$ and all $|\sigma z| < 1$.*

Proof. Since $E_{1,\sigma}(z) = (1 - \sigma z)^{-1}$ for all $|\sigma z| < 1$, Proposition 3.6 implies that $(1 - \sigma z)^{-1} \in \mathcal{P}_1$ for all $|\sigma z| < 1$. Viceversa, let $f(z) = \sum_{p=0}^{\infty} c_p z^p$ be a function in $\mathcal{P}_1$ and let $R_f > 0$ be its radius of convergence. Then it must hold that $c_p = c_1^p$ for all $p \geq 1$ and that $R_f = 1/c_1$. By renaming $c_1 = \sigma$ we get that $f(z) = (1 - \sigma z)^{-1} + c_0 - 1$ for all $|\sigma z| < 1$. By Lemma 3.7, we have that $c_0 = 1$ and therefore the conclusion. □

Remark 3.9. *Proposition 3.8 gives us a characterization of the resolvent, identifying it as the only analytic function in $\mathcal{P}$ that belongs to $\mathcal{P}_1$.*

Notice that, as a consequence of Propositions 3.6 and 3.8, the most popular communicability measures used in network science, namely, the exponential-based and resolvent-based communicability measures, are defined using functions that belong to $\mathcal{P}_\gamma$ for some $\gamma > 0$. However, not all $f \in \mathcal{P}$ belong to $\mathcal{P}_\gamma$ for some $\gamma > 0$. Indeed, it is easy to see for example that power series with positive coefficients that decay as a double exponential, i.e., functions in $\mathcal{P}$ such that $c_p = 1/\exp(\exp(p))$, cannot be in $\mathcal{P}_\gamma$ for any $\gamma \in (0, 1]$.[2] There are also some functions in $\mathcal{P}$, such as $f(z) = 1 + \sum_{p=1}^{\infty} p^{-1} z^p$, that do not even belong to $\mathcal{P}_0$.[3] Some examples are reported in Table 3.2.

[2] For all $\gamma \in (0, 1]$ there exist two $p, q \geq 1$ such that such that $\log(c_{p+q}) = -\exp(p + q) \leq -\exp(p) - \exp(q) - (p + q)\log(1/\gamma) = \log(\gamma^{p+q} c_p c_q)$, hence we have that $c_{p+q} \leq c_p c_q \gamma^{p+q}$. On the other hand, $\log(c_{p+q}) = -\exp(p + q) \leq -\exp(p) - \exp(q) = \log(c_p c_q)$ for all $p, q \geq 1$, which means that $c_{p+q} \leq c_p c_q$ for all $p, q \geq 1$.

[3] In fact, $c_{p+q} = (p + q)^{-1} \geq (pq)^{-1} = c_p c_q$ for all $p, q \geq 2$.

Table 3.2: Some examples of functions in $\mathcal{P}_\gamma$ for some $\gamma \in [0,1]$ and some examples of functions in $\mathcal{P}$ that do not belong to $\mathcal{P}_\gamma$ for any $\gamma \in [0,1]$.

function		γ
resolvent	$E_{0,1}(z) = (1-z)^{-1} = \sum_{p=0}^{\infty} z^p$, $\lvert z\rvert < 1$	1
Mittag-Leffler	$E_{\alpha,\beta}(z) = \sum_{p=0}^{\infty} \frac{\beta^p}{\Gamma(\alpha p+1)} z^p$, $\alpha \in (0,1)$, $\beta > 0$	$1/2^\alpha$
exponential	$E_{1,1}(z) = \exp(z) = \sum_{p=0}^{\infty} \frac{1}{p!} z^p$	$1/2$
example 4	$1 + \sum_{p=1}^{\infty} \frac{1}{\exp(\exp(p))} z^p$	0
example 5	$1 + \sum_{p=1}^{\infty} \frac{1}{p^s} z^p$, $s \geq 1$, $\lvert z\rvert < 1$	$\nexists$

We conclude this section by showing how to construct functions that belong to $\mathcal{P}_\gamma$ using the Hadamard product of power series [61].

Proposition 3.10. *Let $f(z) = \sum_{p=0}^{\infty} c_p z^p$ be such that $f \in \mathcal{P}_\gamma$ for some $\gamma \in [0,1]$, and let $g(z) = \sum_{p=0}^{\infty} c'_p z^p$ be a function in $\mathcal{P}_{\gamma'}$ for some $\gamma' \in [0,1]$. Then, the Hadamard product*

$$f * g(z) := \sum_{p=0}^{\infty} c_p c'_p z^p$$

is such that $f * g \in \mathcal{P}_{\gamma\gamma'}$ for all $\lvert z\rvert < \min\{R_f, R_g\}$.

Proof. If $f, g \in \mathcal{P}$, then $f * g \in \mathcal{P}$ and its radius of convergence its equal to the minimum between the radii of f and g. Now, letting $\hat{c}_p := c_p c'_p$, it is easy to check that $(\gamma\gamma')^{p+q} \hat{c}_p \hat{c}_q \leq \hat{c}_{p+q} \leq \hat{c}_p \hat{c}_q$ for all $p, q \geq 0$. □

Observe that, as a result of Proposition 3.10, we have that $\mathcal{P}_0$ and $\mathcal{P}_1$ are closed under the Hadamard product of power series.

3.3. Counting walks through a set of directed edges

In this section we present formulas for counting the number of r-walks that visit at least one of the elements in $\mathcal{E}$. These expressions will be useful for studying the modification of f-communicability in the downdate and update problems presented in Section 3.1. We recall that, in this chapter, we focus on directed graphs (see Chapter 2 for the undirected case).

Let $H_r^{\mathcal{E}} \in \mathbb{R}^{n\times n}$, $r \geq 0$, be the matrix such that $\mathbf{e}_i^T H_r^{\mathcal{E}} \mathbf{e}_j$ is the number of walks of length $r \geq 0$[4] from i to j that pass through at least one of the (possibly virtual) edges in $\mathcal{E}$. We have the following lemma, which is a direct consequence of Lemma 1.13.

Lemma 3.11. *In the above notation, it holds that*

$$H_r^{\mathcal{E}} = (M_{\mathcal{E}})^r - M^r, \text{ for all } r > 0.$$

The remainder of this section provides some equivalent formulations for $H_r^{\mathcal{E}}$ in terms of $\mathcal{E}$-avoiding walks (Definition 1.10).

Proposition 3.12. *Let $\mathcal{E} = \{e\}$, where $e = (u, v)$ is a (possibly virtual) edge, and let $r > 0$. Then,*

$$H_r^{\mathcal{E}} = \sum_{k=0}^{r-1} M^k \mathbf{e}_u \mathbf{e}_v^T (M_{\mathcal{E}})^{r-k-1}.$$

Before writing the proof, we state the following fact, which follows straightforwardly from Lemma 1.13 and is the fundamental tool that we use to prove the next two results.

Lemma 3.13. *If $\mathcal{E}$ is a set of (possibly virtual) edges, then $\mathbf{e}_i^T M^r \mathbf{e}_j$ is equal to the number of $\mathcal{E}$-avoiding walks of length r from node i to node j.*

Proof of Proposition 3.14. Fix $i, j \in V$ and proceed entrywise. Each r-walk from i to j that visits $e = (u, v) \in \mathcal{E}$ consists of an $\mathcal{E}$-avoiding walk of length k from i to u, followed by e, and then a walk of length $r - k - 1$ from v to j, that may visit e. For each $k = 0, \ldots, r-1$, these can be counted by $\mathbf{e}_i^T M^k \mathbf{e}_u \mathbf{e}_v^T (M_{\mathcal{E}})^{r-k-1} \mathbf{e}_j$ according to Lemma 1.13 and Lemma 3.13. Since every walk is counted exactly once in this way, we get the result. □

The following proposition can be proved using a similar argument.

Proposition 3.14. *Let $\mathcal{E} \subset E$ be a set of (possibly virtual) edges, with $e = (u_e, v_e)$ for each $e \in \mathcal{E}$, and let $r > 0$. Then,*

$$H_r^{\mathcal{E}} = \sum_{e\in\mathcal{E}} \sum_{k=0}^{r-1} M^k \mathbf{e}_{u_e} \mathbf{e}_{v_e}^T (M_{\mathcal{E}})^{r-k-1}. \tag{3.3}$$

[4] If $r = 0$, we simply set $H_0^{\mathcal{E}} = I$.

3.4. Main results

This section contains bounds on the modification of $f(A)_{ij}$ after the removal (resp., addition) of a set or edges (resp., virtual edges) $\mathcal{E}$ to (resp., from) a strongly connected graph[5] with adjacency matrix A. Similar results on the f-based Estrada index and f-based total network communicability are also included as direct consequences of such bounds.

Henceforth, f will denote an analytic function with finite or infinite radius of convergence R_f, that belongs to $\mathcal{P}_\gamma$ for some $\gamma \in [0,1]$, and that satisfies the hypotheses of Theorem 1.19 (which we recall below for easy reference) for the matrix $M_{\mathcal{E}}$.

Theorem 1.19. ([65, Theorem 4.7]) *Let $A \in \mathbb{C}^{n\times n}$ be a matrix with distinct eigenvalues $\lambda_1, \ldots, \lambda_s$ and let f be such that $f(z) = \sum_{p=0}^{\infty} c_p z^p$ for any $|z| < R_f$. Then $f(A)$ is well defined and*

$$f(A) = \sum_{r=0}^{\infty} c_p A^p \tag{1.1}$$

if and only if $\lambda_1, \ldots, \lambda_s$ satisfy one of the following:

1. *$|\lambda_k| < R_f$ for all $k = 1, \ldots, s$;*
2. *$|\lambda_k| = R_f$ and the series for $f^{(n_k-1)}(\lambda)$ is convergent at $\lambda = \lambda_k$, for all $k = 1, \ldots, s$.*

Moreover, we will assume to have *full knowledge* on the graph with adjacency matrix A. This means having access to quantities such as $f(A)$, $f(\gamma A)$, or $f(\gamma^2 A)$ which are treated as if they were computed prior to the graph perturbation. Recall that, when $\mathcal{E} \subset E$, we have that $M_{\mathcal{E}} = A$, whereas $M = A$ holds when $\mathcal{E} \subset \overline{E}$ (see Table 3.1). Lastly, we will require that M is irreducible[6] to ensure that $f(\gamma' M)_{ij} > 0$ for all $0 < \gamma' \leq \gamma$ and all $i, j = 1, \ldots, n$ (see Lemma 3.18 below).

Remark 3.15. *If $f \in \mathcal{P}$ satisfies the hypotheses of Theorem 1.19 for $M_{\mathcal{E}}$, then the same hypotheses hold for M as well (see Remark 1.20).*

[5] If the graph was not strongly connected, all the results would apply straightforwardly to each of its strongly connected component.

[6] This assumption is needed only if $\gamma > 0$ (see Remark 3.20).

3.4.1. Estimating the change in f-communicability

We begin by proving two lemmas that will be useful in this section.

Lemma 3.16. *Let $X, Y \in \mathbb{R}^{n\times n}$ be such that, entrywise, $0 \leq X \leq Y$. Then,*

$$0 \leq f(X)_{ij} \leq f(Y)_{ij} \text{ for all } i,j = 1,\ldots,n,$$

for all $f \in \mathcal{P}$ that satisfies the assumptions of Theorem 1.19 for Y.

Proof. Since $0 \leq X \leq Y$ we have that, entrywise, $0 \leq X^p \leq Y^p$ for each $p \geq 0$ [68, Equation 8.1.12]. Applying (1.1) yields the conclusion since $0 \leq c_p X^p \leq c_p Y^p$ for all $p \geq 0$, where $c_p \geq 0$ is the pth Taylor coefficients of f. □

Remark 3.17. *Lemma 3.16 ensures that $f(M)_{ij} \leq f(M_{\mathcal{E}})_{ij}$ for all $i,j = 1,\ldots,n$ whenever $f \in \mathcal{P}$ satisfies the assumptions of Theorem 1.19 for $M_{\mathcal{E}}$. In other words, we have that the f-communicability is always non-increasing (resp., non-decreasing) with respect to the removal (resp., addition) of a subset of edges from (resp., to) a graph.*

Lemma 3.18. *Let $f \in \mathcal{P}$ satisfy the assumptions of Theorem 1.19 for $X \geq 0$ irreducible $n \times n$ matrix. Then $f(X)_{ij} > 0$ for all $i,j = 1,\ldots,n$.*

Proof. Fix i,j and proceed entrywise. Since $X \geq 0$ is irreducible, there exists $q = q(i,j) > 0$ such that $(X^q)_{ij} > 0$. Since $f \in \mathcal{P}$, we have that, entrywise, $f(X)_{ij} > c_q(X^q)_{ij} > 0$, where c_q is the qth Taylor coefficient of f. □

We are now ready to prove the main result of this chapter.

Theorem 3.19. *Let $\mathcal{E}$ be a subset of (possibly) virtual edges such that $e = (u_e, v_e)$ for all $e \in \mathcal{E}$. Then, for each $i,j = 1,\ldots,n$,*

$$\gamma^2 c_1 \sum_{e\in\mathcal{E}} f(\gamma^2 M)_{iu_e} f(\gamma M)_{v_e j} \leq f(M_{\mathcal{E}})_{ij} - f(M)_{ij} \leq c_1 \sum_{e\in\mathcal{E}} f(M_{\mathcal{E}})_{iu_e} f(M_{\mathcal{E}})_{v_e j}, \tag{3.4}$$

for all $f \in \mathcal{P}_\gamma$, with $\gamma \in [0,1]$ that satisfies the hypothesis of Theorem 1.19 for $M_{\mathcal{E}}$.

Proof. From (1.5) and Lemma 3.11 we have that

$$f(M_{\mathcal{E}}) - f(M) = \sum_{r=1}^{\infty} c_r H_r^{\mathcal{E}}.$$

By plugging (3.3) into the last expression and by interchanging the order of summation we get

$$\begin{aligned} f(M_{\mathcal{E}}) - f(M) &= \sum_{e\in\mathcal{E}} \sum_{r=1}^{\infty} c_r \sum_{k=0}^{r-1} M^k \mathbf{e}_{u_e} \mathbf{e}_{v_e}^T (M_{\mathcal{E}})^{r-k-1} \\ &= \sum_{e\in\mathcal{E}} \sum_{k=0}^{\infty} \sum_{r=k+1}^{\infty} c_r M^k \mathbf{e}_{u_e} \mathbf{e}_{v_e}^T (M_{\mathcal{E}})^{r-k-1} \\ &= \sum_{e\in\mathcal{E}} \sum_{k=0}^{\infty} \sum_{h=0}^{\infty} c_{k+h+1} M^k \mathbf{e}_{u_e} \mathbf{e}_{v_e}^T (M_{\mathcal{E}})^{h}. \end{aligned}$$

The fact that $f \in \mathcal{P}_\gamma$ and Lemma 3.4 yield the following entrywise inequalities:

$$\begin{aligned} f(M_{\mathcal{E}})_{ij} - f(M)_{ij} &\le c_1 \sum_{e\in\mathcal{E}} \sum_{k=0}^{\infty} c_k \mathbf{e}_i^T (M)^k \mathbf{e}_{u_e} \sum_{h=0}^{\infty} c_h \mathbf{e}_{v_e}^T (M_{\mathcal{E}})^h \mathbf{e}_j \\ &= c_1 \sum_{e\in\mathcal{E}} f(M)_{iu_e} f(M_{\mathcal{E}})_{v_e j}; \end{aligned}$$

and

$$\begin{aligned} f(M_{\mathcal{E}})_{ij} - f(M)_{ij} &\ge \gamma^2 c_1 \sum_{e\in\mathcal{E}} \sum_{k=0}^{\infty} \gamma^{2k} c_k \mathbf{e}_i^T M^k \mathbf{e}_{u_e} \sum_{h=0}^{\infty} \gamma^h c_h \mathbf{e}_{v_e}^T (M_{\mathcal{E}})^h \mathbf{e}_j \\ &= \gamma^2 c_1 \sum_{e\in\mathcal{E}} f(\gamma^2 M)_{iu_e} f(\gamma M_{\mathcal{E}})_{v_e j}. \end{aligned}$$

Lemma 3.16 gives us $f(M)_{ij} \le f(M_{\mathcal{E}})_{ij}$ and $f(\gamma M_{\mathcal{E}})_{ij} \ge f(\gamma M)_{ij}$ for each $i, j = 1, \ldots, n$, and therefore the conclusion. □

Remark 3.20. *If $f \in \mathcal{P}_0$ the lower bound in Theorem 3.19 becomes trivial. This observation holds for all the bounds derived in the rest of this chapter.*

Notice that, depending on whether $\mathcal{E} \subset E$ or $\mathcal{E} \subset \overline{E}$, one of the two bounds in (3.4) is cost-free, while the other requires some calculations. In the downdate problem, for instance, we assume to have access to the matrix $f(M_{\mathcal{E}})$ (see Table 3.1). This means that we can obtain the upper bound in (3.4) without any further computational effort. The corresponding lower bound, on the other hand, requires the computation of $|\mathcal{E}|$ columns of $f(\gamma M)$ and $|\mathcal{E}|$ columns of $f(\gamma^2 M)$. Nevertheless, these latter operations are still considerably cheaper than computing the full matrix $f(M)$ (see Section 1.2.1), an operation that would completely defeat the purpose of the bounds presented in this chapter. A similar reasoning holds for the update problem, with the role of the upper and

lower bounds reversed. Furthermore, in practical applications, we are usually interested in just one of the two bounds in (3.4). For example, when assessing the robustness of a network in the downdate problem, an upper bound is more useful than a lower bound as it can represent worst-case scenarios. Conversely, when adding connections to enhance a network in the update problem, a lower bound is more informative, as it helps quantifying the minimum guaranteed increase in communicability.

3.4.2. Bounds on related network indices

We now apply our results to devise upper and lower bounds on the modification of the f-based Estrada index and f-based total network communicability, defined as in (1.6)–(1.7) in Chapter 1. We will rely on these bounds in the experimental section to estimate the effect of edge removal/addition on the connectivity of the whole network. As in the previous section, unless otherwise specified, henceforth f denotes an analytic function that belongs to $\mathcal{P}_\gamma$ for some $\gamma \in [0,1]$ that satisfies the hypotheses of Theorem 1.19 for $M_{\mathcal{E}}$.

We will make use of the following corollary, which can be obtained from Theorem 3.19 when $|\mathcal{E}| = 1$.

Corollary 3.21. *Let $\mathcal{E} = \{e\}$ such that $e = (u, v)$ is a (possibly virtual) edge. Then,*

$$\gamma^2 c_1 f(\gamma^2 M)_{iu} f(\gamma M)_{vj} \leq f(M_{\mathcal{E}})_{ij} - f(M)_{ij} \leq c_1 f(M_{\mathcal{E}})_{iu} f(M_{\mathcal{E}})_{vj}, \qquad (3.5)$$

for all $f \in \mathcal{P}_\gamma$ with $\gamma \in [0,1]$ that satisfies the hypothesis of Theorem 1.19 for $M_{\mathcal{E}}$.

Corollary 3.22. *Let $\mathrm{EE}_f(M_{\mathcal{E}})$ and $\mathrm{EE}_f(M)$ be the f-based Estrada index of the graphs with adjacency matrix $M_{\mathcal{E}}$ and M, respectively. Then,*

$$\frac{\gamma^2 c_1}{n} \mathbf{e}_v^T f(\gamma^2 M) f(\gamma M) \mathbf{e}_u \leq EE_f(M_{\mathcal{E}}) - EE_f(M) \leq \frac{c_1}{n} \mathbf{e}_v^T f(M_{\mathcal{E}}) f(M_{\mathcal{E}}) \mathbf{e}_u. \qquad (3.6)$$

Proof. From the definition of f-based Estrada we get

$$EE_f(M_{\mathcal{E}}) - EE_f(M) = \frac{1}{n} \sum_{i=1}^{n} f(M_{\mathcal{E}})_{ii} - f(M)_{ii}.$$

By using Corollary 3.21, we have

$$\frac{\gamma^2 c_1}{n} \sum_{i=1}^{n} f(\gamma^2 M)_{iu} f(\gamma M)_{vi} \leq EE_f(M_{\mathcal{E}}) - EE_f(M) \leq \frac{c_1}{n} \sum_{i=1}^{n} f(M_{\mathcal{E}})_{iu} f(M_{\mathcal{E}})_{vi},$$

form which we get the conclusion. □

A similar result can be proved for the f-based total network communicability.

Corollary 3.23. *Let* $\mathrm{TC}_f(M_{\mathcal{E}})$ *and* $\mathrm{TC}_f(M)$ *be the* f*-based total network communicability of the graphs with adjacency matrix* $M_{\mathcal{E}}$ *and* M*, respectively. Then,*

$$\frac{\gamma^2 c_1}{n}\mathbf{1}^T f(\gamma^2 M)\mathbf{e}_u\mathbf{e}_v^T f(\gamma M)\mathbf{1} \leq \mathrm{TC}_f(M_{\mathcal{E}})-\mathrm{TC}_f(M) \leq \frac{c_1}{n}\mathbf{1}^T f(M_{\mathcal{E}})\mathbf{e}_u\mathbf{e}_v^T f(M_{\mathcal{E}})\mathbf{1}. \tag{3.7}$$

Proof. From the definition f-based total network communicability we get

$$TC_f(M_{\mathcal{E}}) - TC_f(M) = \frac{1}{n}\sum_{i,j=1}^{n} f(M_{\mathcal{E}})_{ij} - f(M)_{ij}.$$

By using Corollary 3.21, we have

$$\frac{\gamma^2 c_1}{n}\sum_{i,j=1}^{n} f(\gamma^2 M)_{iu} f(\gamma M)_{vj} \leq \mathrm{TC}_f(M_{\mathcal{E}})-\mathrm{TC}_f(M) \leq \frac{c_1}{n}\sum_{i,j=1}^{n} f(M_{\mathcal{E}})_{iu} f(M_{\mathcal{E}})_{vj},$$

from this we get the conclusion. □

Remark 3.24. *Corollary 3.23 suggests that, in order to minimize the effect of edge removal on* f*-total network communicability, one should remove the edge* $e = (u, v)$ *that minimizes the product* $\mathbf{1}^T f(M_{\mathcal{E}})\mathbf{e}_u\mathbf{e}_v^T f(M_{\mathcal{E}})\mathbf{1}$*. This was exactly the heuristic proposed in [5] for the downdate problem on digraphs.*

3.5. Extension to weighted graphs

In this section we extend the results proved in Section 3.4 by turning to the problem of *weight perturbation* in a weighted graph. In a real-life scenario, such perturbations may represent the strengthening or weakening of certain connections in a transportation network, with the purpose of making it more resilient to failures and/or cheaper to maintain.

Mathematically, the weight perturbation problem is also equivalent to a modification of the (real) adjacency matrix of the graph. In this sense, the weight perturbation problem can be seen as a relaxed variant of the downdate and update problems considered in Section 3.1, where modifications were restricted to flipping the entries of a (binary) adjacency matrix between 0 and 1. Indeed, let G be weighted graph (Definition 1.12) with adjacency matrix

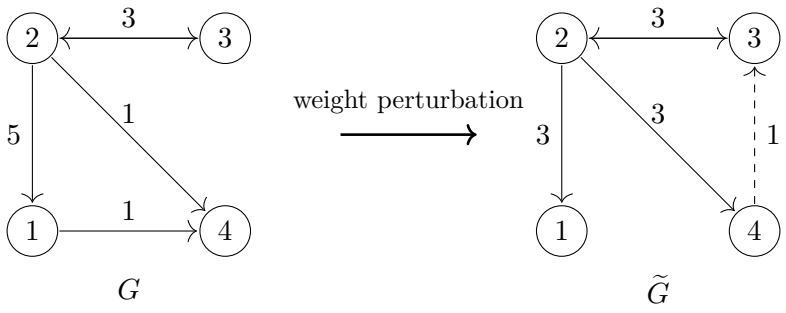


Figure 3.2: Weight perturbation problem on a toy graph with four nodes.

$A = (\omega_{ij}) \in \mathbb{R}^{n\times n}$, $\omega_{ij} \geq 0$, and let $\widetilde{G}$ as the graph obtained from G by perturbing the weights of a set of (possibly virtual) edges $\mathcal{E}$, i.e., the weighted graph with adjacency matrix $\widetilde{A} = (\widetilde{\omega}_{ij}) \in \mathbb{R}^{n\times n}$, where

$$\widetilde{\omega}_{ij} = \begin{cases} \omega_{ij} + \Delta\omega_{ij} & \text{if } (i,j) \in \mathcal{E} \\ \omega_{ij} & \text{otherwise,} \end{cases}, \quad \Delta\omega_{ij} \geq -\omega_{ij},$$

for all $i, j = 1, \ldots, n$. Then, it is easy to see that $\widetilde{\omega}_{ij} \geq 0$ and that

$$\widetilde{A} - A = \sum_{e\in\mathcal{E}} \Delta\omega_{u_e v_e} \mathbf{e}_{u_e} \mathbf{e}_{v_e}^T, \tag{3.8}$$

where $e = (u_e, v_e) \in \mathcal{E}$ are the edges whose weights have been perturbed.

For example, let us consider weight perturbation problem in Figure 3.2, in which the weights of the edges $(2,1)$ and $(2,4)$ in G decrease and increase by two units respectively, and the weights of the edges $(1,4)$ and $(4,3)$ change in such a way that $(1,4)$ and $(4,3)$ are, respectively, removed and added to G. Then, $\Delta\omega_{21} = -2$, $\Delta\omega_{24} = 2$, $\Delta\omega_{14} = -1$, $\Delta\omega_{43} = 1$, and the corresponding adjacency matrices are given by

$$A = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 5 & 0 & 3 & 1 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \text{ and } \tilde{A} = \begin{pmatrix} 0 & 0 & 0 & \mathbf{0} \\ \mathbf{3} & 0 & 3 & \mathbf{3} \\ 0 & 3 & 0 & 0 \\ \mathbf{1} & 0 & 0 & 1 \end{pmatrix}.$$

Remark 3.25. *All $\Delta\omega_{ij}$ can take positive or negative values. Moreover, they can be independent of each other. These facts will play a significant role in some of the results discussed below.*

Now, let G be a weighted graph and let $\widetilde{G}$ be the graph obtained from G by perturbing the weights of a set $\mathcal{E}$ of (possibly virtual) edges. To extend Theorem 3.19, we first need to generalise Proposition 3.14 to the case of weight perturbation. Here, for clarity, we show an algebraic proof of it. However, a combinatorial proof in the spirit of the proof of Proposition 3.14 would also work.

Proposition 3.26. *In the above notation, we have that*

$$\widetilde{A}^r - A^r = \sum_{e\in\mathcal{E}} \Delta\omega_{u_e v_e} \sum_{k=0}^{r-1} \widetilde{A}^k \mathbf{e}_{u_e} \mathbf{e}_{v_e}^T A^{r-k-1},\ r \geq 1.$$

Proof. Using [67, Chapter 6, Problem 7], we have

$$\widetilde{A}^r - A^r = \sum_{k=0}^{r-1} \widetilde{A}^k (\tilde{A} - A) A^{r-k-1},\ r \geq 1.$$

Plugging (3.8) into the last equation yields the conclusion. □

We can now proceed to prove the following theorem.

Theorem 3.27. *Let $f \in \mathcal{P}_\gamma$ for some $\gamma \in [0,1]$ that satisfies the hypotheses of Theorem 1.19 for A and $\tilde{A}$. Then, for all $i,j = 1,\dots,n$, we have*

$$\begin{aligned} f(\tilde{A})_{ij} - f(A)_{ij} \leq\, & c_1 \sum_{e\in\mathcal{E}:\Delta\omega_{u_e v_e}>0} \Delta\omega_{u_e v_e} f(\tilde{A})_{i u_e} f(A)_{v_e j} \\ & + \gamma^2 c_1 \sum_{e\in\mathcal{E}:\Delta\omega_{u_e v_e}<0} \Delta\omega_{u_e v_e} f(\gamma^2\tilde{A})_{i u_e} f(\gamma A)_{v_e j}, \end{aligned}$$

and

$$\begin{aligned} f(\tilde{A})_{ij} - f(A)_{ij} \geq\, & c_1 \sum_{e\in\mathcal{E}:\Delta\omega_{u_e v_e}<0} \Delta\omega_{u_e v_e} f(\tilde{A})_{i u_e} f(A)_{v_e j} \\ & + \gamma^2 c_1 \sum_{e\in\mathcal{E}:\Delta\omega_{u_e v_e}>0} \Delta\omega_{u_e v_e} f(\gamma^2\tilde{A})_{i u_e} f(\gamma A)_{v_e j}, \end{aligned}$$

where $e = (u_e, v_e) \in \mathcal{E}$ are the edges whose weight has been perturbed.

Proof. We show the proof for the upper bound, as the proof for the lower bound follows accordingly. From (1.5) we have that

$$f(\tilde{A}) - f(A) = \sum_{r=1}^{\infty} c_r (\widetilde{A}^r - A^r).$$

By using Proposition 3.26 and interchanging the order of summation we get

$$f(\tilde{A}) - f(A) = \sum_{e\in\mathcal{E}} \Delta\omega_{u_e v_e} \sum_{k=0}^{\infty}\sum_{h=0}^{\infty} c_{k+h+1} \widetilde{A}^k \mathbf{e}_{u_e} \mathbf{e}_{v_e}^T A^h.$$

Now, by splitting the summation according to the sign of $\Delta\omega_{u_e v_e}$ and using Lemma 3.4, we get the following entrywise inequality

$$\begin{aligned} f(\tilde{A})_{ij} - f(A)_{ij} \leq c_1 \sum_{e\in\mathcal{E}:\Delta\omega_{u_e v_e}>0} \Delta\omega_{u_e v_e} \sum_{k=0}^{\infty} c_k \mathbf{e}_i^T \widetilde{A}^k \mathbf{e}_{u_e} \sum_{h=0}^{\infty} c_h \mathbf{e}_{v_e}^T A^h \mathbf{e}_j \\ + \gamma^2 c_1 \sum_{e\in\mathcal{E}:\Delta\omega_{u_e v_e}<0} \Delta\omega_{u_e v_e} \sum_{k=0}^{\infty} \gamma^{2k} c_k \mathbf{e}_i^T \widetilde{A}^k \mathbf{e}_{u_e} \sum_{h=0}^{\infty} \gamma^h c_h \mathbf{e}_{v_e}^T A^h \mathbf{e}_j, \end{aligned}$$

from which we get the conclusion by applying (1.5). □

Theorem 3.27 can be considered as a generalization of Theorem 3.19 to the case of weight perturbation. Indeed, it is easy to see that Theorem 3.27 reduces to Theorem 3.19 if G is unweighted and $\Delta\omega_{u_e v_e} = \pm 1$ for all $e = (u_e, v_e) \in \mathcal{E}$.

Remark 3.28. *In the statement of Theorem 3.27, f must satisfy the hypotheses of Theorem 1.19 for both A and $\tilde{A}$ since, in general, we do not known if $\rho(A) > \rho(\tilde{A})$ or $\rho(\tilde{A}) > \rho(A)$.*

Corollary 3.29. *In the above notation, if $\mathcal{E} = \{(u, v)\}$ and $\Delta\omega_{uv} > 0$ it holds*

$$\gamma^2 c_1 \Delta\omega_{uv} f(\gamma^2 A)_{iu} f(\gamma A)_{vj} \leq f(\tilde{A})_{ij} - f(A)_{ij} \leq c_1 \Delta\omega_{uv} f(\tilde{A})_{iu} f(\tilde{A})_{vj}.$$

Similarly, if $\Delta\omega_{uv} < 0$

$$c_1 \Delta\omega_{uv} f(\tilde{A})_{iu} f(\tilde{A})_{vj} \leq f(\tilde{A})_{ij} - f(A)_{ij} \leq \gamma^2 c_1 \Delta\omega_{uv} f(\gamma^2 A)_{iu} f(\gamma A)_{vj}.$$

Proof. The proof follows by applying Theorem 3.27 and using the fact that, as a consequence of Lemma 3.16, $f(A)_{iu} \leq f(\tilde{A})_{iu}$ when $\Delta\omega_{uv} > 0$, while $f(A)_{iu} \leq f(\tilde{A})_{iu}$ when $\Delta\omega_{uv} < 0$. □

Remark 3.30. *If $\Delta\omega_{uv} \approx 0$, i.e.,if the weight perturbation applied to (u, v) is close to zero we have that, according to Corollary 3.29, $f(\tilde{A})_{ij} \approx f(A)_{ij}$ for all $i, j = 1, \ldots, n$. This confirms the intuition that a "small perturbation" of the graph induces a small perturbation in the values of f-communicability.*

Now we extend the bounds on the f-based Estrada index and the f-based total network communicability devised in Section 3.4 to the weighted case. Both proofs are omitted since they are essentially the same as the proof of Corollaries 3.22 and 3.23, respectively.

Corollary 3.31. *In the notation of Corollary 3.29, if $\Delta\omega_{uv} > 0$, we have*

$$\frac{\gamma^2 c_1 \Delta\omega_{uv}}{n}\mathbf{e}_v^T f(\gamma^2 A) f(\gamma A)\mathbf{e}_u \leq EE_f(\tilde{A}) - EE_f(A) \leq \frac{c_1\Delta\omega_{uv}}{n}\mathbf{e}_v^T f(\tilde{A}) f(\tilde{A})\mathbf{e}_u. \tag{3.9}$$

If $\Delta\omega_{uv} < 0$, the roles of upper and lower bounds are reversed.

Corollary 3.32. *In the notation of Corollary 3.29, if $\Delta\omega_{uv} > 0$, we have*

$$\frac{\gamma^2 c_1 \Delta\omega_{uv}}{n}\mathbf{1}^T f(\gamma^2 A)\mathbf{e}_u\mathbf{e}_v^T f(\gamma A)\mathbf{1} \leq \mathrm{TC}_f(\tilde{A}) - \mathrm{TC}_f(A) \leq \frac{c_1\Delta\omega_{uv}}{n}\mathbf{1}^T f(\tilde{A})\mathbf{e}_u\mathbf{e}_v^T f(\tilde{A})\mathbf{1}. \tag{3.10}$$

If $\Delta\omega_{uv} < 0$, the roles of upper and lower bounds are reversed.

3.6. Numerical examples

In this section we report some preliminary results on the bounds obtained in Sections 3.4 and 3.5 on some synthetic and real-world networks. All experiments are carried out in MATLAB R2023b on a PC equipped with an Intel i7-1260P CPU with a clock rate of 2.10 GHz, and 16 GB of RAM.

The first example we consider is the synthetic graph from [91] displayed in Figure 3.3, consisting of two undirected cycles with eleven nodes each, connected by a non-reciprocated edge between node 11 and node 12. In this experiment, we examine the case in which the virtual edge $e = (12, 11)$ is added to the graph, and we compare the bounds in (3.5) with the true change in f-communicability. The results are presented in Figure 3.4, where we show the bounds in (3.5) alongside the actual value of $f(M_{\mathcal{E}})_{ii} - f(M)_{ii}$, which we computed using MATLAB's `expm` in the case of $f(z) = \exp(z)$ and MATLAB's `inv` in the case of $f(z) = r_{1/3}(z) = (1 - z/3)^{-1}$. In this figure, we also show the upper bound from [91] with $\epsilon = 0.03$ for $r_{1/3}(z)$ (no parameters are needed for $\exp(z)$).

The results indicate that our bounds in (3.5) successfully reproduce the qualitative behaviour of $f(M_{\mathcal{E}})_{ii} - f(M)_{ii}$ for all $i = 1, \ldots, n$. In particular, both the upper and lower bounds reveal that the nodes most affected by the addition of edge $e = (12, 11)$ are the closest, in terms of graph distance, to nodes 11 and 12. These findings are in line with what observed in [91] for both $f(z) = \exp(z)$ and $f(z) = r_{1/3}(z) = (1 - z/3)^{-1}$. Moreover, we see that the upper bound in (3.5) is sharper than the one in [91] for the nodes i associated with the largest change in f-communicability, while it is looser for other values corresponding to the smallest changes. This suggest that the

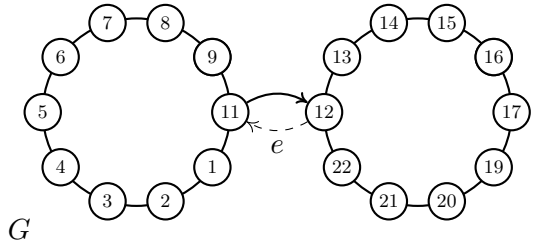


Figure 3.3: A synthetic graph that consists in two undirected cycles, with eleven nodes each, connected by a direct edge between node 11 and 12. In our experiment, we consider the case in which the virtual edge $e = (12, 11)$, represented by a dashed line in the figure, is added to the graph.

two bounds are not strictly comparable in terms of tightness in this particular example. Nevertheless, the lower bounds in Figure 3.4 are perhaps the most informative quantities in this example, as they quantify the minimum possible increase in f-communicability resulting from the addition edge $e = (12, 11)$.

In this next series of experiments, we test the bounds in (3.6)–(3.10) by examining how the f-based Estrada index and the f-based total network communicability change when a perturbation affects some edges in a (possibly weighted) graph. In particular, we consider two scenarios:

- the removal of 15 randomly selected edges from the directed an unweighted `gnutella08` network; and
- the weight increase (by increments of varying magnitude) in a random edge in the undirected and weighted `sandi-authors` network.

Both examples are real-world networks taken from the SuiteSparse Matrix collection [36]. Specifically, the `gnutella08` network belongs to the SNAP group and represents a snapshot of the Gnutella file sharing network from the 8 August 2002. Its largest strongly connected component has 2068 nodes and 9313 edges. The `sandi-authors` collaboration network belongs to the Pajek group and its largest connected component consists of 86 nodes and 124 weighted edges. Since the weights in the `sandi-authors` network are integer numbers, we rescale the weights to ensure that $\omega_{ij} \in [0, 1]$ for all $i, j = 1, \ldots, n$. By doing this, we make sure that the weight of the perturbed edge (which is chosen at random) is modified in a meaningful way when $\Delta\omega$ varies between 10^{-1} and 10^{-10}. For both experiments, all reported values represent an average taken from 20 runs with the same parameters.

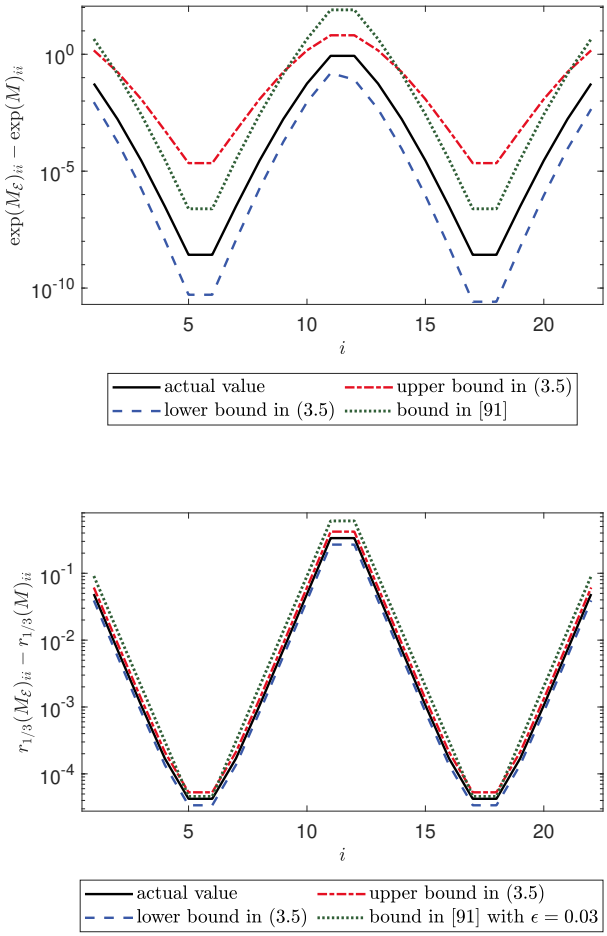


Figure 3.4: Comparisons between the bounds (3.5), the upper bound in [91] with $\epsilon = 0.03$ for $r_{1/3}(z)$ (no parameters are needed for $\exp(z)$), and the actual value of $f(M_{\mathcal{E}})_{ii} - f(M)_{ii}$ after the addition of the edge $e = (12, 11)$ to the graph in Figure 3.3. In the plots, f is chosen to be either $f(z) = \exp(z)$ (top figure) or $f(z) = r_{1/3}(z) = (1 - z/3)^{-1}$ with (bottom figure).

The results are reported in Figures 3.5 and 3.6. We see that the bounds in (3.6)–(3.7) and (3.9)–(3.10) are able to reproduce the variation of $\mathrm{EE}_f(A)$ and $\mathrm{TC}_f(A)$ (although sometimes only qualitatively, as in the case of $f(z) = \exp(z)$), for both the networks. In particular, the experiment on the `gnutella08` network (Figure 3.5) shows that the bounds in (3.6)–(3.7) remain accurate even when the perturbations affects multiple edges in sequence. At the same time, the experiment on the `sandi-authors` network shows that small perturbation of the weights (i.e., small values of $\Delta\omega$ in Figure 3.6) are associated to small variation of the f-based Estrada index and the f-based total network communicability, as expected.

3.7. Conclusions and outlook

In this chapter we introduced a special class of analytic functions $\mathcal{P}_\gamma$ whose Taylor coefficients are positive and satisfy $\gamma^{p+q} c_p c_q \leq c_{p+q} \leq c_p c_q$, for all $p, q \geq 0$ and some $\gamma \in [0, 1]$. We also showed that all Mittag-Leffler functions, which include the exponential function, belong to $\mathcal{P}_\gamma$ for certain $\gamma \in (0, 1]$ and that the resolvent is the only analytic function that belongs to $\mathcal{P}_1$.

After proving a result that enumerates all walks visiting a set of directed edges using the concept of $\mathcal{E}$-avoiding walks, and thus extending some of the result from Chapter 2, we derived upper and lower bounds on the changes in f-communicability caused by perturbing (adding or deleting) a set of edges in a graph, where $f \in \mathcal{P}_\gamma$ for some $\gamma \in [0, 1]$. These results were then extended to the more general case of weight perturbation in weighted digraphs.

We concluded this chapter with preliminary numerical experiments to validate our bounds and to compare them with existing ones in the literature. The results for $f(z) = \exp(z)$ and $f(z) = (1-z)^{-1}$ showed that our bounds are able to qualitatively reproduce the changes in f-communicability and related quantities after edge perturbation in all tested configurations. Comparison with previous bounds on $f(A)_{ii}$ showed that the sharpness of our bounds alternates depending on node i.

As mentioned in the introduction of this chapter, its content is part of a work in preparation. Further numerical experiments on Mittag–Leffler functions and a study of how the bounds depend on the choice of parameters are in pipeline.

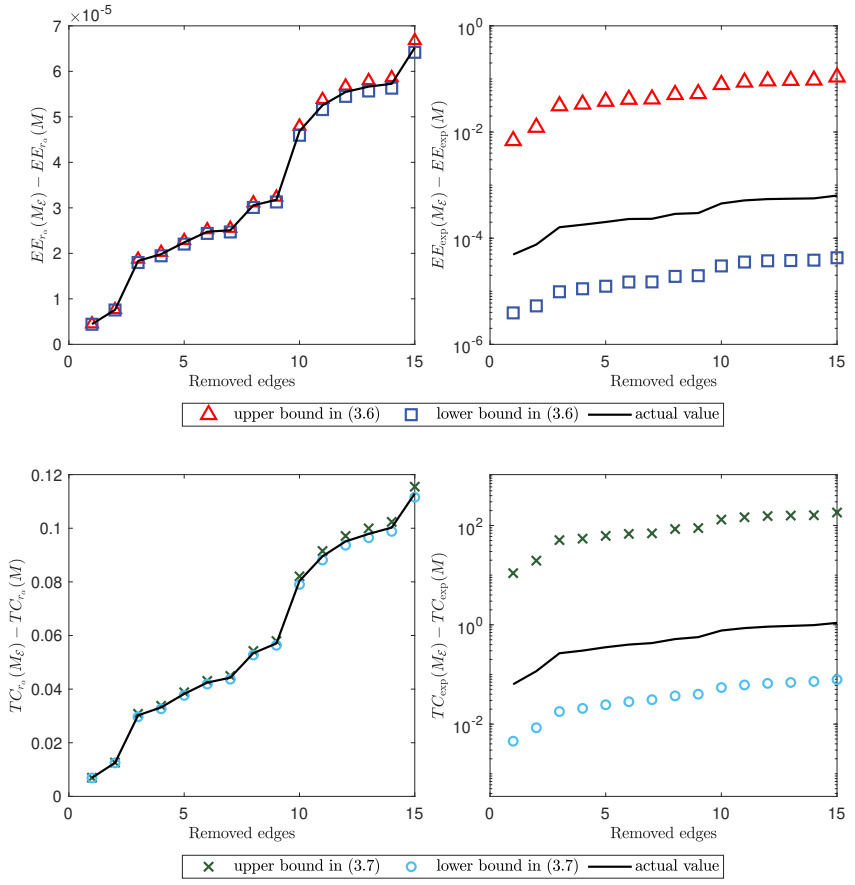


Figure 3.5: Bounds (3.6)–(3.7) compared with the actual value of $\mathrm{EE}_f(M_{\mathcal{E}}) - \mathrm{EE}_f(M)$ and $\mathrm{TC}_f(M_{\mathcal{E}}) - \mathrm{TC}_f(M)$ after the sequential removal of 15 edges (chosen uniformly at random) from the `gnutella08` network. Here f is chosen to be either $f(z) = \exp(z)$ (top right and bottom right figures) or $f(z) = r_\alpha(z) = (1 - \alpha z)^{-1}$ with $\alpha = 0.85/\rho(A)$ (top left and bottom left figures). All the values are averaged over 20 runs with the same parameters.

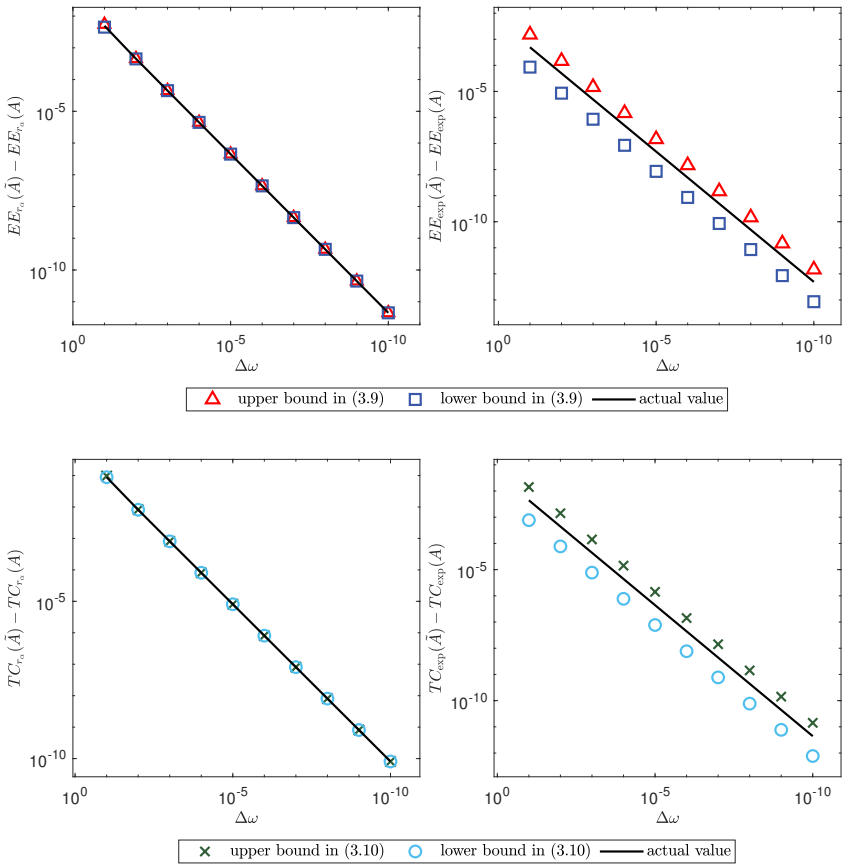


Figure 3.6: Comparisons between the bounds (3.9)–(3.10) and the actual variation of $\mathrm{EE}_f(A)$ and $\mathrm{TC}_f(A)$ after the addition of $\Delta\omega$ to the weight of an existing edge (chosen uniformly at random) of the `sandi-authors` network. Here, f is chosen to be either $f(z) = \exp(z)$ (top right and bottom right figures) or $f(z) = r_\alpha(z) = (1 - \alpha z)^{-1}$, with $\alpha = 0.85/\rho(A)$ (top left and bottom left figures). All the values are averaged over 20 runs with the same parameters.

4. A new walk-based centrality measure for acyclic digraphs

Many real-life cascade processes can be described as source-to-sink walks in a directed acyclic network (Definition 1.11), where the source node represents the triggering event and the sink node represents the final outcome. In such acyclic networks, makes sense to measure the importance of a node by counting the number of source-to-sink walks that visit it. In fact, the higher the number of these walks that pass through a node, the more cascade reactions will be affected by its removal from the network.

In this chapter, whose results were recently presented in [24], we propose *source-to-sink communicability*, a novel walk-based centrality measure to rank nodes in a directed acyclic graph (DAG), along with an efficient way of computing and updating it after the removal of an edge from the graph. We also discuss a possible application to the analysis of some biological pathways.

4.1. Some centrality measures for DAGs

As already mentioned in Chapter 1, a centrality measure is typically defined as a function that assigns a nonnegative value to each node, quantifying its importance within a network [44, 87]. In the case of directed networks, however, it is common practice to give each node two different centrality values, one measuring the ability of the node to spread information and one measuring its capabilities as an information receiver [20, 75].

Perhaps the most simple way to measure the importance of a node $w \in V$ in a directed network is to use its in-degree $\mathrm{ind}(w)$ and its out-degree $\mathrm{outd}(w)$ (see Definition 1.4). Other famous examples are *hub centrality* $\mathbf{h}$ and *authority centrality* $\mathbf{a}$ that are obtained by the *hyperlink-induced topic search* (HITS) algorithm [75], and the *PageRank* $\boldsymbol{\pi}$ and *Reverse PageRank* $\boldsymbol{\pi}^{(rev)}$, that are defined as the Perron eigenvector of two stochastic matrices obtained from A and A^T (see [54] for more details). Regarding the class of centrality measures

based on walk enumeration [47], *Katz broadcast* and *Katz receiver centralities* [73, 22] are among the most popular. They can be written as the solutions of two linear systems involving A and A^T respectively, i.e.,

$$(I - \alpha A)\mathbf{k}^{(b)} = \mathbf{1} \quad \text{and} \quad (I - \alpha A^T)\mathbf{k}^{(r)} = \mathbf{1},$$

where α is a parameter chosen such that $0 < \alpha < 1/\rho(A)$, with $\rho(A)$ being the spectral radius of A. A detailed analysis on the detection of hubs and authorities through matrix functions, such as the matrix exponential, can be found in [20].

In many applications, such as the analysis of biological networks, the interest often lies in identifying nodes whose removal significantly affects the functionality of the network [71]. This issue typically involves simulating the targeted removal of a set of nodes according to a given centrality measure; see, e.g., [27, 69] and references therein. To this extent, it can be useful to devise a unique score that combines the broadcast and receiving centrality of a node, thus summarising its importance in a directed network. In view of this need, the authors of [83] heuristically suggest using the sum of broadcast and receiver Katz centrality, or the sum of PageRank and Reverse PageRank, i.e. twice their average, calling it "source-sink centrality". In Section 4.4 and 4.5, we provide comparisons with these indices, denoting them by $\boldsymbol{\pi} + \boldsymbol{\pi}^{(rev)}$ and $\mathbf{k}^{(b)} + \mathbf{k}^{(r)}$ to distinguish them from our proposal.

4.2. Walk counting in a DAG

In this section we derive a formula for counting walks that visit a node in a directed acyclic graph, which forms the basis of the new centrality measure introduced in this chapter.

We begin by recalling some well-known properties of DAGs.

Theorem 4.1 (see, e.g., [30])**.** *Let G be a directed acyclic graph, and let $A \in \mathbb{R}^{n\times n}$ be its adjacency matrix, then the following hold:*

(a) Every walk in G is a path.

(b) The concatenation[1] of two paths in G, one from i to w, and one from w to j always results in a path from i to j that visits w.

(c) G has at least one node with in-degree zero (a source), and at least one node with out-degree zero (a sink).

[1] We recall that the *concatenation* of an r-walk $i_0, \ldots, i_{r-1}, w$ and an ℓ-walk $w, \ldots, j_{\ell-1}, j_\ell$ is the walk $i_0, \ldots, i_{r-1}, w, j_1, \ldots, j_\ell$ which has a length of $r + \ell$.

(d) There exists a permutation matrix P such that $\tilde{A} = P^{-1}AP$ is upper triangular with zeros on the diagonal and therefore nilpotent[2]*.*

Theorem 4.1 allow us to easily enumerate r-walks, and therefore source-to-sink walks of length r that visit a specific node $w \in V$. In fact, we can state the following result.

Proposition 4.2. *Let A be the adjacency matrix of an acyclic digraph, then for each node $w \in V$,*

$$H_r^w = \sum_{k=0}^{r} A^k \mathbf{e}_w \mathbf{e}_w^T A^{r-k}, \ r > 0, \tag{4.1}$$

is the matrix such that, entrywise, $\mathbf{e}_i^T H_r^w \mathbf{e}_j$ is the number of r-walks from i to j that visit w.

Proof. Consider three nodes i, j, and w, then for all $r > 0$ it holds that

$$\mathbf{e}_i^T H_r^w \mathbf{e}_j = \begin{cases} 0 & \text{if } i = w = j \\ \mathbf{e}_i^T A^r \mathbf{e}_j & \text{if } i = w \neq j \text{ or } i \neq w = j \\ \mathbf{e}_i^T A^k \mathbf{e}_w \mathbf{e}_w^T A^{r-k} \mathbf{e}_j & \text{if } i \neq w \neq j \end{cases}.$$

Since $\mathbf{e}_w^T A^r \mathbf{e}_w = 0$ for all $r > 0$, and since $\mathbf{e}_i^T A^r \mathbf{e}_j$ equals the number of r-walks from i to j, we are left to prove the result only in the case of $i \neq w \neq j$. Indeed, if $i \neq w \neq j$, we know from (a) and (b) in Theorem 4.1 that each r-walk from i to j that visit w is the concatenation of a path of length k from i to w, that can be counted by $\mathbf{e}_i^T A^k \mathbf{e}_w$, and a path of length $r - k$ from w to j, that can be counted by $\mathbf{e}_w^T A^{r-k} \mathbf{e}_j$. Thus, for any given $0 \leq k \leq r$, the quantity $\mathbf{e}_i^T A^k \mathbf{e}_w \mathbf{e}_w^T A^{r-k} \mathbf{e}_j$ is equal to the number of r-walks from i to j that visit w at step k. Summing over all $k = 0, \ldots, r$ yields the result. □

Let η_r^w the number of source-to-sink walks of length $r > 0$ that visit node w, then an immediate consequence of Proposition 4.2 is the following:

$$\eta_r^w = \sum_{\substack{i\,:\,\mathrm{ind}(i)=0 \\ j\,:\,\mathrm{outd}(j)=0}} \mathbf{e}_i^T H_r^w \mathbf{e}_j$$

or, for short,

$$\eta_r^w = \mathbf{s}^T H_r^w \mathbf{t},$$

[2] A is said to be nilpotent if $A^k = 0$ for some positive integer k.

where $\mathbf{s}, \mathbf{t}$ are vectors in $\mathbb{R}^n$ such that

$$(\mathbf{s})_i := \begin{cases} 1 & \text{if } \operatorname{ind}(i) = 0 \\ 0 & \text{otherwise} \end{cases}, \quad (\mathbf{t})_j := \begin{cases} 1 & \text{if } \operatorname{outd}(j) = 0 \\ 0 & \text{otherwise} \end{cases}.$$

Moreover, letting $\bar{\eta}$ be the total number of source-to-sink walks in a DAG with adjacency matrix A, the next proposition holds.

Proposition 4.3. *In the above notation,*

$$\bar{\eta} = \mathbf{s}^T (I - A)^{-1} \mathbf{t}. \tag{4.2}$$

Proof. For Theorem 4.1 we have that A is nilpotent and, consequently,

$$\sum_{l=0}^{\infty} \mathbf{s}^T A^l \mathbf{t} = \sum_{l=0}^{n-1} = \mathbf{s}^T A^l \mathbf{t} = \mathbf{s}^T (I - A)^{-1} \mathbf{t}.$$

Lemma 1.13 and the definitions of $\mathbf{s}$ and $\mathbf{t}$ yield the conclusion. □

4.3. A new centrality score for DAGs

In this section we present a new centrality index built specifically for DAGs that can be computed by solving two linear systems involving $(I-A)$, $(I-A^T)$, and the vectors $\mathbf{s}$ and $\mathbf{t}$.

4.3.1. Source-to-sink communicability, stsc

Let G be an acyclic digraph without isolated nodes (Definition 1.5) and let $A \in \mathbb{R}^{n \times n}$ be its adjacency matrix. We propose to measure the centrality of a node $w \in V$ with the fraction of source-to-sink walks that visit it, i.e., the quantity

$$x_w = \frac{1}{\bar{\eta}} \sum_{r=1}^{\infty} \eta_r^w = \frac{1}{\bar{\eta}} \sum_{r=1}^{\infty} \mathbf{s}^T H_r^w \mathbf{t}, \tag{4.3}$$

where η_r^w is the the number of source-to-sink walks of length r that visit w, and $\bar{\eta}$ is the overall number of source-to-sink walks in G.

Observe that, as a consequence of (d) in Theorem 4.1, A is a nilpotent matrix, which implies the convergence of the series since there exists a $r^* > 0$ such that $H_r^w = 0$ for each $r \geq r^*$. Moreover, (c) in Theorem 4.1 and the absence of isolated nodes in G ensure that $\bar{\eta} > 0$ and that $x_w > 0$ for all $w \in V$.

We call $\bar{\eta} = \mathbf{s}^T(I-A)^{-1}\mathbf{t}$ the *total source-to-sink communicability* of a DAG with adjacency matrix A. We choose this name to resemble the *(resolvent based) total network communicability* introduced by Benzi and Klymko [21], which is instead defined as $TC = \mathbf{1}^T(1-\alpha A)^{-1}\mathbf{1}$[3], with the parameter α chosen such that $0 < \alpha < 1/\rho(A)$. Both quantities, TC and $\bar{\eta}$, rely on the enumeration of walks in G to measure the general connectivity of a network. However, TC was initially proposed for undirected graphs and considers all walks that connect each pair of nodes, while $\bar{\eta}$ counts only source-to-sink walks, making it specific to the context of DAGs. Similarly to what the authors do in [21], we refer to the centrality score x_w as the *source-to-sink communicability* (for short, *stsc*) of node $w \in V$.

Proposition 4.4. *In the above notation,*

$$x_w = \frac{1}{\bar{\eta}}\left[\mathbf{s}^T(I-A)^{-1}\mathbf{e}_w\mathbf{e}_w^T(I-A)^{-1}\mathbf{t}\right], \quad \textit{for all } w = 1\ldots,n. \tag{4.4}$$

Proof. By plugging (4.1) into (4.3), and by rearranging the terms in the sums we get

$$\begin{aligned}
\bar{\eta}x_w &= \sum_{r=1}^{\infty}\sum_{k=0}^{r}(\mathbf{s}^TA^k\mathbf{e}_w)(\mathbf{e}_w^TA^{r-k}\mathbf{t}) \\
&= \sum_{k=0}^{\infty}\sum_{r=k}^{\infty}(\mathbf{s}^TA^k\mathbf{e}_w)(\mathbf{e}_w^TA^{r-k}\mathbf{t}) - \mathbf{s}^T\mathbf{e}_w\mathbf{e}_w^T\mathbf{t} \\
&= \sum_{k=0}^{\infty}\mathbf{s}^TA^k\mathbf{e}_w\sum_{h=0}^{\infty}\mathbf{e}_w^TA^h\mathbf{t} - \mathbf{s}^T\mathbf{e}_w\mathbf{e}_w^T\mathbf{t}
\end{aligned}$$

By Theorem 4.1, A is nilpotent, $(I-A)$ is invertible, and

$$\sum_{l=0}^{\infty}A^l = \sum_{l=0}^{n}A^l = (I-A)^{-1}.$$

Therefore,

$$\bar{\eta}x_w = \mathbf{s}^T(I-A)^{-1}\mathbf{e}_w\mathbf{e}_w^T(I-A)^{-1}\mathbf{t} - \mathbf{s}^T\mathbf{e}_w\mathbf{e}_w^T\mathbf{t}.$$

The result yields from the fact that $\mathbf{s}^T\mathbf{e}_w\mathbf{e}_w^T\mathbf{t} = 0$ for all nodes w that are not isolated. □

In some applications, it could be useful to assign length-dependent weights to all source-to-sink walks when computing the *stsc* centrality score of a node.

[3] TC is usually defined with a scale factor of $1/n$ to compare different sized networks.

Specifically, one could be interested in penalising long-range source-to-sink interactions, as they could be less crucial for the overall network connectivity [46]. In practice, this can be done by substituting A with βA in Proposition 4.4, where $0 < \beta \leq 1$ is an attenuation parameter that controls the importance given to shorter walks compared to longer ones, so that each walk of length $r > 0$ is weighted by β^r.

Definition 4.5. *Let A be the adjacency matrix of a directed acyclic network, and let* $\mathbf{s}$ *and* $\mathbf{t}$ *be as in Section 4.2. The* parametric source-to-sink communicability of node $w = 1, \ldots, n$ *is given by*

$$x_w^\beta = \frac{1}{\mathbf{s}^T(I - \beta A)^{-1}\mathbf{t}} \left[\mathbf{s}^T(I - \beta A)^{-1}\mathbf{e}_w\mathbf{e}_w^T(I - \beta A)^{-1}\mathbf{t}\right], \; 0 < \beta \leq 1.$$

For the remainder of Section 4.3, we focus on the case $\beta = 1$ to simplify the presentation. However, the discussion can be extended to include *parametric stsc* as well.

4.3.2. Computing stsc

Let $\mathbf{y} = (y_w)$ and $\mathbf{z} = (z_w)$ be the unique solutions of the following linear systems[4] :

$$(I - A^T)\mathbf{y} = \mathbf{s}, \text{ and} \tag{4.5}$$

$$(I - A)\mathbf{z} = \mathbf{t}, \tag{4.6}$$

then for (4.4) we have that, entrywise, $\bar{\eta}x_w = y_w z_w$ for all $w = 1, \ldots, n$. Moreover, as a consequence of (4.2), $\bar{\eta} = \mathbf{s}^T\mathbf{z}$. Thus, to compute the vector $\mathbf{x} = (x_w)$ such that x_w is the source-to-sink communicability of $w \in V$, we need to solve two linear systems of order n, compute a scalar product in $\mathbb{R}^n$, and perform one entrywise product between two vectors of $\mathbb{R}^n$.

Therefore, efficiently solving the two linear systems in (4.5) and (4.6) is of paramount importance, particularly when n is large and/or several DAGs have to be considered. In particular, the two linear systems can be transformed into one upper and one lower triangular because A can be transformed into an upper triangular nilpotent matrix $\tilde{A}$, see (d) in Theorem 4.1, and solved using a direct method.

Proposition 4.6. *Let us suppose that the upper and lower triangular linear systems (4.5) and (4.6) have bandwidth d, possibly after applying a reordering. Then, the underlying linear systems can be solved in $O(nd)$ operations, where*

[4]Recall that for (d) in Theorem 4.1 $(I - A)$ and $(I - A^T)$ are nonsingular.

d is the maximum number of nonzero diagonals of the underlying triangular matrices.

Proof. Follows by applying a Reverse Cuthill-McKee algorithm to a symmetrization of the underlying triangular matrices (RCMs for short), then applying banded back substitution and forward elimination algorithm that cost $O(nd)$ operations; see, e.g., [56, Section 4.3]. By supposing that the computational cost of RCMs does not impact more than $O(nd)$ flops, we get the result. □

Since these linear systems are completely independent, they can be solved at the same time, e.g., in a multicore CPU system. From moderate to large n, this direct solution strategy could not be the most appropriate or even unsuitable if n is large or very large in a parallel environment. In this case, an iterative Krylov subspace solver can be better suited. A review of these iterative solvers and their usage can be found in [25].

We stress that all experiments in Sections 4.4 and 4.5 use the direct solution since all networks tested have relatively small dimension.

4.3.3. Update

In several applications, one may wish to study how centrality indices change with respect to low-rank modifications of the adjacency matrix, such as the removal or the addition of an edge to the network. For example, in the analysis of biological pathways, it can be useful to consider a case where a mutation disrupts specific protein interactions within the network. In this context, having a method for updating a centrality measure can be quite beneficial, as it may help reduce the computational cost of recalculating the indices repeatedly from scratch.

The goal of this section is to present a formula for updating the *stsc* vector $\mathbf{x}$ after removing[5] an edge from an acyclic digraph G. Assuming that G is topologically sorted, the proposed formula can be significantly more convenient than recomputing the *stsc* scores from scratch (see Proposition 4.8 for more details).

Proposition 4.7. *Let G be an acyclic digraph with $\bar{\eta} > 1$, and let A be its adjacency matrix. Moreover, let $\hat{\mathbf{x}} = (\hat{x}_w)$ be the vector such that $\hat{x}_w$ is the source-to-sink communicability of node $w = 1, \ldots, n$ in the graph obtained from*

[5] Although we concentrate on edge removal, the discussion can be extended to also include edge addition.

G by removing the edge (u, v). Then, if $\mathrm{ind}(v), \mathrm{outd}(u) > 1$,

$$\hat{x}_w = \frac{1}{\bar{\eta} - y_u z_v} \left[y_w z_w - y_w \mathbf{e}_w^T (I - A)^{-1} \mathbf{e}_u z_v - y_u \mathbf{e}_v^T (I - A)^{-1} \mathbf{e}_w z_w \right], \quad (4.7)$$

where $\mathbf{y} = (y_w)$ and $\mathbf{z} = (z_w)$ are vectors such that $y_w = \mathbf{e}_w^T (I - A^T)^{-1} \mathbf{s}$, and $z_w = \mathbf{e}_w^T (I - A)^{-1} \mathbf{t}$.

Proof. Given that (u, v) is an edge in G, there can be no walks from v to u. In fact, if such a walk existed, there would be a cycle in G. Hence,

$$\mathbf{e}_v^T (I - A)^{-1} \mathbf{e}_u = \sum_{k=0}^{\infty} \mathbf{e}_v^T A^k \mathbf{e}_u = 0.$$

Consequently, the Sherman-Morrison formula [96] applied to $(I - \hat{A})^{-1}$, where $\hat{A} := A - \mathbf{e}_u \mathbf{e}_v^T$ is the adjacency matrix of the graph obtained from G by removing (u, v), yields the following equation

$$(I - \hat{A})^{-1} = (I - A)^{-1} - (I - A)^{-1} \mathbf{e}_u \mathbf{e}_v^T (I - A)^{-1}.$$

Since $\mathrm{ind}(v), \mathrm{outd}(u) > 1$, the removal of (u, v) does not introduce new sources or sinks in the graph. Therefore, removing (u, v) from G leaves $\mathbf{s}$ and $\mathbf{t}$ unchanged and we have that, for each $w = 1, \ldots, n$,

$$\begin{aligned} \mathbf{s}^T (I - \hat{A})^{-1} \mathbf{e}_w \mathbf{e}_w^T (I - \hat{A})^{-1} \mathbf{t} = {} & y_w z_w + y_u \mathbf{e}_v^T (I - A)^{-1} \mathbf{e}_w \mathbf{e}_w^T (I - A)^{-1} \mathbf{e}_u z_v \\ & - y_w \mathbf{e}_w^T (I - A)^{-1} \mathbf{e}_u z_v - y_u \mathbf{e}_v^T (I - A)^{-1} \mathbf{e}_w \mathbf{e}_v z_w, \end{aligned}$$

where $\mathbf{y} = (y_w)$ and $\mathbf{z} = (z_w)$ are the vectors $\mathbf{y} = (I - A^T)^{-1} \mathbf{s}$ and $\mathbf{z} = (I - A)^{-1} \mathbf{t}$. In addition, by using (4.2) we have that

$$\mathbf{s}^T (I - \hat{A})^{-1} \mathbf{t} = \bar{\eta} - y_u z_v,$$

which is positive since $\bar{\eta} > 1$. The observation that $\mathbf{e}_v^T (I - A)^{-1} \mathbf{e}_w \mathbf{e}_w^T (I - A)^{-1} \mathbf{e}_u = 0$ for each $w = 1, \ldots, n$ (otherwise there would be a cycle in G), as well as Propositions 4.3 and 4.4, yields the conclusion. □

Now, let us assume that G is topologically sorted (see Definition 1.2), this means that A is upper-triangular with zeros on the diagonal and can be written blockwise as

$$A = \begin{bmatrix} A_u & \cdots & \cdots \\ 0 & A_{v-u-1} & \cdots \\ 0 & 0 & A_{n-v+1} \end{bmatrix}, \text{for each } 1 \leq u < v \leq n,$$

where $A_u \in \mathbb{R}^{u\times u}$, $A_{v-u-1} \in \mathbb{R}^{v-u-1\times v-u-1}$, and $A_{n-v+1} \in \mathbb{R}^{n-v+1\times n-v+1}$ are upper-triangular matrices with zeros on the diagonal too. It is well known that $(I-A)^{-1}$ exists if and only if it can be written as

$$(I-A)^{-1} = \begin{bmatrix} (I_u - A_u)^{-1} & \cdots & \cdots \\ 0 & (I_{v-u-1} - A_{v-u-1})^{-1} & \cdots \\ 0 & 0 & (I_{n-v+1} - A_{n-v+1})^{-1} \end{bmatrix},$$

where I_u, I_{v-u-1}, and I_{n-v+1} are identity matrices of appropriate sizes. As a consequence,

$$\mathbf{e}_w^T (I-A)^{-1} \mathbf{e}_u = \begin{cases} [(I_u - A_u)^{-1}]_{w,u} & \text{if } w = 1,\ldots,u \\ 0 & \text{otherwise,} \end{cases}$$

and

$$\mathbf{e}_v^T (I-A)^{-1} \mathbf{e}_w = \begin{cases} [(I_{n-v+1} - A_{n-v+1})^{-1}]_{1,w-v+1} & \text{if } w = v,\ldots,n \\ 0 & \text{otherwise,} \end{cases}$$

which implies that (4.7) in Proposition 4.7 can be rewritten in compact form as

$$\hat{\mathbf{x}} = \frac{1}{\bar{\eta} - y_u z_v} \left[\mathbf{y} \circ \mathbf{z} - \begin{pmatrix} \mathbf{y}_{1:u} \circ z_v \mathbf{p} \\ \mathbf{0} \\ \mathbf{z}_{v:n} \circ y_u \mathbf{q} \end{pmatrix} \right], \tag{4.8}$$

where $\circ$ denotes the elementwise multiplication of vectors, $\mathbf{y}_{1:u}$ and $\mathbf{z}_{v:n}$ denote, respectively, the vectors that contain the first u elements of $\mathbf{y}$ and the last v elements of $\mathbf{z}$, $\mathbf{0} \in \mathbb{R}^{v-u-1}$, and $\mathbf{p} \in \mathbb{R}^u$ and $\mathbf{q} \in \mathbb{R}^{n-v+1}$ are the solutions of the two non-singular linear systems

$$(I_u - A_u)\mathbf{p} = [0,\ldots,0,1]^T, \text{ and } (I_{n-v+1} - A_{n-v+1}^T)\mathbf{q} = [1,0,\ldots,0]^T. \tag{4.9}$$

We are now to the point of stating the following proposition regarding the computational cost of our update formula.

Proposition 4.8. *Assume that G is topologically sorted, that* $\mathrm{ind}(v), \mathrm{outd}(u) > 1$*, and that $\bar{\eta} > 1$, $\mathbf{y} \in \mathbb{R}^n$, $\mathbf{z} \in \mathbb{R}^n$ are already computed. Computing $\hat{\mathbf{x}}$, the updated* stsc *vector after removing (u,v) from G, using (4.8) requires solving two upper-triangular linear systems of order n_u and n_v, where $n_u + n_v \leq n$, plus $2(n + n_u + n_v)$ multiplications.*

Proof. Since $\bar{\eta}$, $\mathbf{y}$, and $\mathbf{z}$ are assumed to be known, the only unknowns in (4.8) are the vectors $\mathbf{p}$ and $\mathbf{q}$. These two vectors can be computed as the unique

solutions of the two linear systems in (4.9), which are of order u and $n-v+1$, respectively. Let $n_u := u$ and $n_v := n-v+1$, then applying (4.8) requires the solution of two triangular linear systems of order n_u and n_v, three entrywise products that require $n + n_u + n_v$ multiplications, and $n + n_u + n_v$ scalar multiplications. □

Remark 4.9. *Depending on the removed edge, n_u and n_v in Proposition 4.8 can be such that $n_u + n_v \ll n$. For example, if G has n nodes and the removed edge is $(5, n-4)$, then $n_u = n_v = 5$ independently of the value of n.*

4.4. Numerical experiments

All experiments in this section were run in MATLAB (version R2023b) on a laptop equipped with 16 GB of RAM and an Intel i7-1260P CPU with a clock rate of 2.10 GHz.

4.4.1. Tests on toy DAGs

Here we compute source-to-sink communicability for some toy DAGs and we compare it with other centrality scores. The examples are built to emphasize three crucial features of the newly introduced centrality index: *non-locality*, *effective bridge recognition*, and *interpretability*.

Following the proposal in [83], we consider the sum of the broadcast and receiving centrality to obtain a unified centrality score that we compare with source-to-sink communicability. Considering this, we focus on the following measures:

- $\mathbf{d}^{(in)} + \mathbf{d}^{(out)}$, the sum of in-degree and out-degree;
- $\boldsymbol{\pi} + \boldsymbol{\pi}^{(rev)}$, the sum of the PageRank and the Reversed PageRank;
- $\mathbf{k}^{(b)} + \mathbf{k}^{(r)}$, the sum of broadcast and receiver Katz centrality with $\alpha = 0.85$[6];
- $\mathbf{h} + \mathbf{a}$, the sum of hub and authority centrality; and
- $stsc(\beta)$, parametric source-to-sink communicability with $\beta = 1$ and $\beta = 0.8$.

[6]Recall that if A is the adjacency matrix of a DAG then $\rho(A) = 0$, which means that Katz broadcast and receiving centrality are well defined for each parameter $\alpha > 0$.

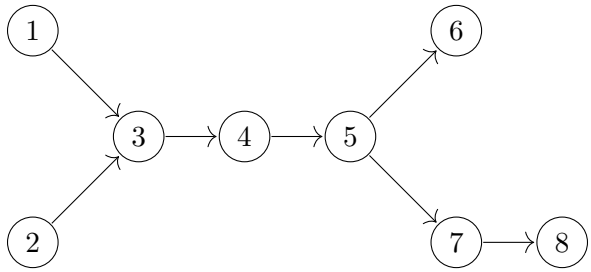


Figure 4.1: Toy network from Example 1

Table 4.1: Centrality scores from Example 1

Node	$\mathbf{d}^{(in)} + \mathbf{d}^{(out)}$	$\boldsymbol{\pi} + \boldsymbol{\pi}^{(rev)}$	$\mathbf{k}^{(b)} + \mathbf{k}^{(r)}$	$\mathbf{h} + \mathbf{a}$	$stsc(1)$	$stsc(0.8)$
1	1	0.183	5.674	0.333	0.500	0.500
2	1	0.183	5.674	0.333	0.500	0.500
3	3	0.337	7.023	0.500	1.000	1.000
4	2	0.347	7.204	< 0.001	1.000	1.000
5	3	0.349	7.223	0.333	1.000	1.000
6	1	0.176	5.231	0.250	0.500	0.556
7	2	0.217	6.081	0.250	0.500	0.444
8	1	0.206	5.596	< 0.001	0.500	0.444

Example 1

Consider the network in Figure 4.1. This example presents a network with eight nodes in which three nodes, 3, 4, and 5, are clearly the most important since they act as "bridges" between the two halves of the network (the subgraphs given by $\{1; 2; 3\}$ and $\{5; 6; 7; 8\}$). Therefore, we expect 3, 4, and 5 to get the highest scores among the nodes.

However, by looking at the centrality scores in Table 4.1, we can see that $\mathbf{h} + \mathbf{a}$ totally fails to classify most of the "bridge nodes" since node 4 is ranked as the least important. Similarly, degree centrality does not identify 4 as one of the most crucial nodes, since $\mathbf{d}^{(in)} + \mathbf{d}^{(out)}$ assigns to node 4 the same score as node 7. Katz centrality, PageRank, $stsc(1)$ and $stsc(0.8)$ correctly classify 3, 4, and 5 as the most important nodes of the DAG, showing a less local nature and good bridge identification capabilities.

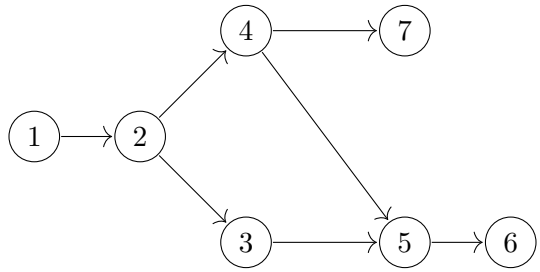


Figure 4.2: Toy network from Example 2

Table 4.2: Centrality scores from Example 2

Node	$\mathbf{d}^{(in)} + \mathbf{d}^{(out)}$	$\boldsymbol{\pi} + \boldsymbol{\pi}^{(rev)}$	$\mathbf{k}^{(b)} + \mathbf{k}^{(r)}$	$\mathbf{h} + \mathbf{a}$	$stsc(1)$	$stsc(0.8)$
1	1	0.345	7.181	< 0.001	1.000	1.000
2	3	0.385	7.946	< 0.001	1.000	1.000
3	2	0.216	5.145	0.382	0.333	0.308
4	3	0.263	5.995	0.618	0.667	0.692
5	3	0.317	7.223	0.618	0.667	0.615
6	1	0.304	6.567	< 0.001	0.667	0.615
7	1	0.171	4.187	0.382	0.333	0.385

Example 2

Now let us consider the network depicted in Figure 4.2. This example is built to test which measure is successful in discriminating between nodes 2, 3, 4 and 5, which all act as "bridges" in the network, although with some differences. In fact, we expect 2 to be ranked as the most important node in the network, as its removal completely shuts down all the paths that connect the single source (node 1) to the two sink nodes (6 and 7). Moreover, we expect 3 to be ranked below 4 and 5, as node 3 appears to be somewhat redundant in the network.

Observing the results shown in Table 4.2, it is possible to see that degree centrality and hubs and authority centrality are the only measures that do not recognise 2 as the most important node in the network. PageRank, Katz centrality, $stsc(1)$, and $stsc(0.8)$ all assign 4 and 5 a higher score than 3, demonstrating effectiveness in determining node hierarchies even within networks where multiple nodes act as "bridges".

Notice that $stsc(0.8)$ is the only score that ranks node 4 higher than node 5 while $stsc(1)$ does not distinguish them. This is largely expected, as node 4 is traversed by shorter source-to-sink walks compared to node 5.

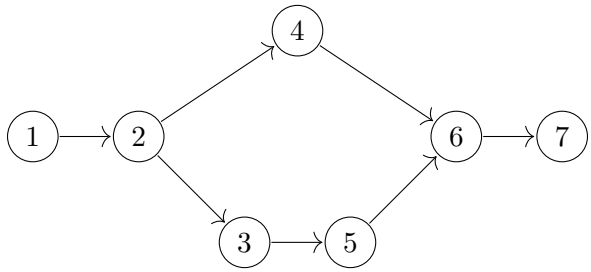


Figure 4.3: Toy network form Example 3

Example 3

The last example we present is the one shown in Figure 4.3. This DAG represents a two-stream network from node 1 (single source) to node 7 (single sink) that can follow two paths: one through 4 and one through 3 and 5. Notice that the path that passes through 4 is shorter than the path that includes 3 and 5. We expect 2 and 6 to get the highest scores in the network as they act as "crossroads" where the two paths diverge and later rejoin. Moreover, we expect little to no differences in the scores of 3, 4, and 5, since the removal of one of these three nodes would result in the disruption of half of the paths connecting 1 with 7.

Looking at the data shown in Table 4.3, we can see that all measures (with the exception of hub and authority centrality) classify nodes 2 and 6 as the most important and simultaneously assign similar scores to nodes 3, 4 and 5. However, few remarks on the influence of the parameter β on $stsc(\beta)$ are in order. First, the scores produced by $stsc(1)$ have the advantage of being easier to interpret compared to those given by $stsc(0.8)$. In fact, the score of 0.500 assigned to nodes 3, 4, and 5 by $stsc(1)$ reflects the fact that removing any of them would break exactly half of the source-to-sink walks in the graph. On the other hand, the scores given by $stsc(1)$ cannot distinguish between the nodes that participate in the shorter path (node 4) and those that belong to the longer one (3 and 5). Depending on the context, one of these characteristics (clear interpretability or sensitivity to walk length) may be more significant than the other, and the choice between $\beta = 1$ or $0 < \beta < 1$ should be made accordingly.

4.4.2. Tests on larger DAGs

We investigated the computational times for *stsc* when considering a series of networks with increasing sizes. The examples are obtained by extracting

Table 4.3: Centrality scores from Example 3

Node	$\mathbf{d}^{(in)} + \mathbf{d}^{(out)}$	$\boldsymbol{\pi} + \boldsymbol{\pi}^{(rev)}$	$\mathbf{k}^{(b)} + \mathbf{k}^{(r)}$	$\mathbf{h} + \mathbf{a}$	$stsc(1)$	$stsc(0.8)$
1	1	0.321	7.011	< 0.001	1.000	1.000
2	3	0.351	7.745	0.333	1.000	1.000
3	2	0.232	5.759	0.250	0.500	0.444
4	2	0.192	5.145	0.583	0.500	0.556
5	2	0.232	5.759	0.333	0.500	0.444
6	3	0.351	7.745	0.500	1.000	1.000
7	1	0.321	7.011	< 0.001	1.000	1.000

the upper triangular part of the symmetric adjacency matrices given by the `erdrey(n,4n)` function in the CONTEST toolbox [64]. This approach yields a topologically sorted DAG. However, by applying a random permutation, we can generate acyclic networks in which the adjacency matrix is not necessarily upper triangular.

The results are illustrated in Figure 4.4, where we examine the average CPU times to calculate *stsc* after the removal/failure of an edge (chosen uniformly at random) from both topologically sorted and unsorted DAGs. For the case of DAGs with upper triangular adjacency matrices, we also report the timings for (4.8), the update formula described in Section 4.3.3. All the reported timings are based on MATLAB's Gaussian elimination.

The data suggest that the computation of *stsc* can be efficient regardless of whether G is topologically sorted or not, as the computational times are relatively low, even for large unsorted DAGs. Moreover, the results show that our update formula for *stsc* can be highly advantageous if the adjacency matrix of the DAG is upper triangular. This is because the linear systems required to apply (4.8) usually have a significantly smaller size than those needed to compute the centrality measure from scratch, which are of order n. In our experiments, we observed that the average sizes of the linear systems solved to apply the update formula (n_u and n_v in Proposition 4.8) were up to 70% smaller than the number of nodes n.

4.5. Application to protein interaction networks

In this section we discuss how source-to-sink communicability can be used to unveil vital nodes in some protein interaction networks that can be modelled as a DAG and we provide comparisons with other popular centrality measures.

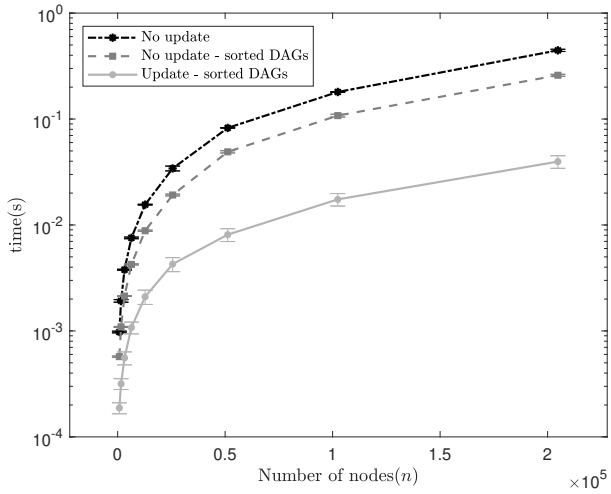


Figure 4.4: Timings (in seconds) to compute the *stsc* scores for some synthetic DAGs with increasing sizes after the removal/failure of a randomly chosen edge. The two dashed lines represent, respectively, the results obtained with the computational scheme described in Section 4.3.2 in the case of sorted and unsorted DAGs. The continuos line shows the results obtained with the update formula presented in Section 4.3.3. The latter is tested only in the case of topologically sorted DAGs. All the values are based on MATLAB's Gaussian elimination and are averaged over 50 runs with the same parameters. Error bars represent standard deviation

4.5.1. Protein interaction networks

Biological systems, such as cells, are the result of a vast array of molecular components that interact with each other at different levels [4]. These components, such as genes, transcripts, proteins, and metabolites, can be schematically depicted as nodes in a complex network, where the interactions between them are illustrated by the edges connecting the nodes [14]. Understanding the way in which the simplest molecular components interact with each other in the cell is one of the hardest steps in systems biology. Thus, some bioinformatics databases, such as STRING [97] or KEGG [72], were built to collect graph representations of various cellular functions, such as signaling, gene regulation, and metabolic processes. The STRING database collects known and predicted interactions between proteins retrieved from the genomic context, experiments, coexpression, text mining, and other databases. However, STRING data do not take into account the complex topology of the pathways [83, 81, 77]. KEGG, instead, contains the representation of signaling and disease-related pathways. In this context, interactions among nodes are assumed to be directed and networks are, mostly, acyclic [78], which means that no path can pass through the same node twice. Few exceptions are made for regulatory motifs and feedback events or metabolic pathways [3]. Therefore, we can generally represent biological pathways as *source-to-sink walks* in a DAG (see Definition 1.11).

In recent years, several tools derived from graph theory and network science have been shown to be useful in extracting information from such protein interaction networks. In particular, centrality measures were proved to be useful for the identification of proteins whose alteration causes the disruption of most cascade reactions; see, e.g., [62, 53, 71, 83]. For example, a recent study [83] heuristically suggest using the sum of broadcast and receiver Katz centrality, or the sum of PageRank and Reverse PageRank, i.e. twice their average, calling it "source-sink centrality". This approach was devised to reflect how the upstream and downstream pathways affect a node, which can be crucial to get a reliable ranking in the context of DAGs modelling biological pathways.

In the following experiments, we study four DAGs that represent some crucial biological pathways we extrapolate from the KEGG database. All the algorithms and the data used in this section, including a Matlab implementation of the `stsc` function that computes the source-to-sink communicability scores, are available on the GitHub repository `https://github.com/alelippo/stsc`.

4.5.2. Pathways description

We choose four pathways from the KEGG database, which contains the representation of several signaling and disease-related pathways. In detail, we selected the pathways with the highest number of nodes that simultaneously have the highest diameter, defined as the maximum path length in the underlying digraph. In line with these requirements, the pathways used in our tests are the following: pathways of neurodegeneration - multiple diseases (`hsa05022`), pathways in cancer (`hsa05200`), Alzheimer's disease (`hsa05010`), and MAPK signaling pathway (`hsa04010`).

- `hsa05022`. Includes the most studied neurodegenerative disorders such as Alzheimer's disease, Parkinson's disease, or Amyotrophic Lateral Sclerosis. It focuses on proteins that are known to be mutated at the genetic level and structures such as the endoplasmic reticulum and mitochondria.
- `hsa05200`. Focuses on typical hallmarks of all cancer diseases, such as uncontrolled proliferation, evading apoptosis, sustained angiogenesis, genomic instability, block of differentiation, resistance to chemotherapy, genomic damage, insensitivity to antigrowth signals, tissue invasion, and metastasis.
- `hsa05010`. Describes key features of Alzheimer's disease such as the accumulation of amyloid beta plaques and neurofibrillary tangles, which are associated with pathological processes such calcium imbalance, synaptic dysfunction, and mitochondrial damage.
- `hsa04010`. It outlines the MAPK signaling pathway that regulates proliferation, differentiation, and inflammation, which are crucial for the progression of the cell cycle and the apoptosis event.

4.5.3. Pathways preprocessing

The preprocessing consists of six main steps: loading, IDs conversion, cleaning, removing duplicates, alias conversion, and adjacency matrix extraction. Using the `keggGet` function of the KEGGREST library [98], the pathways are retrieved from the KEGG database in XML format (eXtensible Markup Language), which provides a structured and machine-readable representation of the pathway data. Since each node is representative of a protein that can be characterized by more than one gene, all gene IDs associated with each protein are retrieved. Then, given that a protein can be included in a pathway in more than one node (this is done in the KEGG database for readability purposes),

Table 4.4: Description of the networks obtained from the pathway preprocessing

Network	n	m	$< \mathbf{d} >$	diam
`hsa05022`	156	195	1.25	11
`hsa05200`	226	304	1.35	17
`hsa05010`	114	118	1.04	9
`hsa04010`	112	155	1.38	10

all nodes made of the same gene IDs are considered as single nodes, and all incoming and outgoing edges are reorganized. Once all nodes are collapsed, the gene IDs are converted to the alias gene name, which is useful for understanding the biological role of the nodes. As a final step, we removed all isolated nodes, reduced all non acyclic digraphs to their *condensation*[7], and extracted the adjacency matrix from the graph.

The network data, including the number of nodes n, the number of edges m, the average degree $< \mathbf{d} >$, and the diameter diam, resulting from the pathway preprocessing, can be found in Table 4.4.

4.5.4. Analysis and comparisons

To show how source-to-sink communicability can be used to extract new informations about the networks in the dataset, we compare the rankings obtained with *stsc*[8] with those given by the centrality scores described in Section 4.4.

One way to compare two node rankings is to look at their correlation, see, e.g., [21, 32]. Indeed, given two rankings $\boldsymbol{\beta}$, we can compute their correlation with the Kendall coefficient κ_τ [74],

$$\kappa_\tau(\boldsymbol{\beta}, \boldsymbol{\gamma}) = \frac{2\,(\mathcal{C} - \mathcal{D})}{n\,(n-1)},$$

where $\mathcal{C}$ is the number of matching pairs and $\mathcal{D}$ is the number of unmatching pairs between the two lists. It is easy to see that $\kappa_\tau \in [-1, 1]$. If κ_τ is close to 1, then the two centrality measures give almost the same rankings on that

[7] We recall that the condensation of G is the graph G^* which has the strongly connected components of G as its nodes, with an edge connecting s_u and s_v whenever there is at least one edge in G from a node in s_u to node in s_v. If G^* is the condensation of a digraph G, then it is easy to prove by contradiction that G^* is acyclic [30].

[8] We focus on the case where $\beta = 1$ since we decided to not take into account the length of the pathways for this specific application.

Table 4.5: Kendall correlation coefficient between the rankings given by *stsc* and those given by the other centrality measures described in Section 4.4

Network	$\mathbf{d}^{(in)} + \mathbf{d}^{(out)}$	κ_τ $\boldsymbol{\pi} + \boldsymbol{\pi}^{(rev)}$	$\mathbf{k}^{(b)} + \mathbf{k}^{(r)}$	$\mathbf{h} + \mathbf{a}$
`hsa05022`	0.01	0.09	0.13	0.00
`hsa05200`	0.03	0.20	0.26	0.01
`hsa05010`	0.13	0.05	0.17	0.16
`hsa04010`	0.07	0.09	0.06	0.04

particular network; if κ_τ is close to -1, then $\boldsymbol{\beta}$ is almost identical to $\boldsymbol{\gamma}^{-1}$, the list that can be obtained by reversing $\boldsymbol{\gamma}$.

Analysing the data in Table 4.5, it becomes evident that *stsc* exhibits a weak correlation with all the measures evaluated, as indicated by the relatively low κ_τ values for each network in the dataset. However, it should be noted that the rankings given by *stsc* appear to be weakly correlated with those given by $\mathbf{d}^{(in)} + \mathbf{d}^{(out)}$, $\mathbf{k}^{(b)} + \mathbf{k}^{(r)}$ and $\mathbf{h} + \mathbf{a}$ in the `hsa05010` network, and with the one given by $\boldsymbol{\pi} + \boldsymbol{\pi}^{(rev)}$ and $\mathbf{k}^{(b)} + \mathbf{k}^{(r)}$ in the `hsa05200` network. These are, respectively, the networks with the lowest and the highest diameters in the dataset. Although it is well known that for the so-called *small world* networks some measures tend to give almost the same rankings of degree centrality, it is significant that, in networks with larger diameter, *stsc* produces rankings that differ significantly from those given by $\mathbf{d}^{(in)} + \mathbf{d}^{(out)}$. In fact, this observation is in line with what can be observed with other non-local measures such as the PageRank and Katz centrality.

Our analysis will now focus on the top 10 highest raked nodes according to each centrality measure. This approach allows us to determine whether *stsc* can be used to reveal crucial nodes that the other centrality indices tested may have overlooked or not ranked appropriately.

In the following, node names that consist of a sequence of elements, such as "Aβ40;Aβ42", are abbreviated by listing only one item. For example, "Aβ40;Aβ42" is abbreviated by "Aβ40". This is done for readability purposes, and since no item is shared among the node names, each abbreviation uniquely identifies a node. A complete list of node names for each pathway is available on the GithHub repository.

Table 4.6: Top 10 nodes of the `hsa05022` network ranked using *stsc* and the other centrality measures described in Section 4.4

$\mathbf{d}^{(in)} + \mathbf{d}^{(out)}$	$\boldsymbol{\pi} + \boldsymbol{\pi}^{(rev)}$	$\mathbf{k}^{(b)} + \mathbf{k}^{(r)}$	$\mathbf{h} + \mathbf{a}$	*stsc*
Aβ40	Aβ40	Aβ40	Aβ40	ATF6
HTT	CASP3	CASP3	HTT	Aβ40
ATF6	AGER	HTT	PRNP	PI3P
NFKB1	PI3P	PRNP	SOD1	CASP3
BECN2	CYCS	KLC3	PSMA8	ERN1
PI3P	CASP9	PI3P	PSMD14	GSK3B
SOD1	NFKB1	Ca^{2+}	SNCA	AGER
CYCS	HTT	C9orf72	GRIN1	MAP3K5
PRNP	O_2^-	CYCS	ATF6	PRNP
GSK3B	ATF6	WIPI2	CACNA1B	KLC3

hsa05022

In this network, most of the centrality measures agree on the importance of "Aβ40" by ranking it as the most crucial node in the pathway. This node represents the *Amyloid beta protein*, that specifically plays a destructive role in Alzheimer's disease forming toxic plaques that disrupt brain cells function and trigger inflammation. Interestingly, *stsc* ranks "Aβ40" as the second most important node. Instead, *stsc* assigns the highest score to "ATF6", a node that represents two proteins involved in the homeostasis of endoplasmic reticulum lowering the level of stress, typical of neurodegenerative diseases. Except for $\mathbf{d}^{(in)} + \mathbf{d}^{(out)}$, which places it as third, the other centrality measures rank "ATF6" significantly lower than what *stsc* does.

Overall, we observe minimal overlap between the rankings produced by the tested measures. A notable exception is given by "CASP3", a crucial executioner caspase of the apoptotic process, which is ranked as the second most important node by both $\boldsymbol{\pi} + \boldsymbol{\pi}^{(rev)}$ and $\mathbf{k}^{(b)} + \mathbf{k}^{(r)}$, and as the fourth node by *stsc*. This is largely expected, as it is known that neuronal death mediated by apoptosis is a common characteristic of several neurodegenerative diseases.

The rankings are reported in Table 4.6.

hsa05200

In this network, there is almost no consensus among the measures regarding the top ten most important nodes. In fact, *stsc* is the only index that ranks "HRAS", "PLCG1", and "AKT3", three key signaling proteins mediated by cytokine-cytokine receptor interaction related to proliferation, evading apop-

Table 4.7: Top 10 nodes of the `hsa05200` network ranked using *stsc* and the other centrality measures described in Section 4.4

$\mathbf{d}^{(in)} + \mathbf{d}^{(out)}$	$\boldsymbol{\pi} + \boldsymbol{\pi}^{(rev)}$	$\mathbf{k}^{(b)} + \mathbf{k}^{(r)}$	$\mathbf{h} + \mathbf{a}$	*stsc*
MYC	AKT3	RB1	MYC	HRAS
EPAS1	HRAS	E2F1	CCND1	PLCG1
HRAS	PIP3	CCND1	VEGFD	AKT3
PIK3R3	PIK3R3	ALK	ETS1	PIP3
AKT3	EPAS1	IGF1R	FOS	PIK3R3
CCND1	CYCS	PDGFRA	JUN	Ca^{2+}
KEAP1	RB1	VEGFD	CXCL8	MAPK1
JAK1	MAPK1	EGFR	MMP1	ARAF
GRB2	E2F1	DCC	CDK4	MAP2K1
NFKB1	GLI1	PLCG1	KEAP1	EGFR

tosis, and sustained angiogenesis through oncogenes, as the top three nodes in the pathway. However, "HRAS" and "AKT3" also receive high scores from $\mathbf{d}^{(in)} + \mathbf{d}^{(out)}$ and $\boldsymbol{\pi} + \boldsymbol{\pi}^{(rev)}$, whereas "PLCG1" receives significantly lower scores from each tested measure, except for *stsc*.

Overall, the rankings given by *stsc* show the greatest similarity to those given by $\boldsymbol{\pi} + \boldsymbol{\pi}^{(rev)}$. For example, the nodes "PIP3" and "PIK3R3" represent, along with "AKT3", three cardinal molecules of the PI3K-AKT signaling pathway that, when out of control, promote cell survival and proliferation in cancerogenic cells. These nodes are ranked third, fourth, and fifth by *stsc* and first, third, and fourth by $\boldsymbol{\pi} + \boldsymbol{\pi}^{(rev)}$. This highlights a preference for this signaling pathway by both stsc and PageRank. In contrast, *stsc* does not include "MYC" or "RB1" among its top ten nodes, despite "MYC" being ranked first by both $\mathbf{d}^{,(in)} + \mathbf{d}^{,(out)}$ and $\mathbf{h} + \mathbf{a}$, and "RB1" being ranked first by $\mathbf{k}^{(b)} + \mathbf{k}^{(r)}$.

The rankings are reported in Table 4.7.

has05010

In this network, *stsc* assigns the highest score to "MAPT". This gene encodes the *tau protein*, which, when abnormally phosphorylated, forms neurofibrillary tangles, typical markers of Alzheimer's disease, that disrupt neuronal function and contribute to the progression of the disease. Notably, $\boldsymbol{\pi} + \boldsymbol{\pi}^{(rev)}$ and $\mathbf{k}^{(b)} + \mathbf{k}^{(r)}$ rank "MAPT" as the second most important node in the network, and $\mathbf{d}^{(in)} + \mathbf{d}^{(out)}$ classifies it as the third most important.

We also notice that *stsc* includes "GSK3B", "TUBA1B" and "TUBB3" among the top 10 most important nodes. These are pivotal genes that encode

Table 4.8: Top 10 nodes of the `hsa05010` network ranked using *stsc* and the other centrality measures described in Section 4.4

$\mathbf{d}^{(in)} + \mathbf{d}^{(out)}$	$\boldsymbol{\pi} + \boldsymbol{\pi}^{(rev)}$	$\mathbf{k}^{(b)} + \mathbf{k}^{(r)}$	$\mathbf{h} + \mathbf{a}$	*stsc*
Aβ40	Aβ40	Aβ40	Aβ40	MAPT
GSK3B	MAPT	MAPT	PI3P	GSK3B
MAPT	GSK3B	TUBA1B	AGER	CASP3
NFKB1	CASP3	TUBB3	MAP3K5	MAPK1
PI3P	IRS1	NDUFA4L2	COX6B2	MAP2K1
PSEN1	PSEN1	Ca^{2+}	CHRM1	Aβ40
CASP3	AGER	CASP3	GRIN1	NDUFA4L2
Ca^{2+}	NFKB1	MAP2K1	CACNA1C	Ca^{2+}
IRS1	Ca^{2+}	MAPK1	HSD17B10	TUBA1B
PPP3CA	PIK3R3	GSK3B	FAS	TUBB3

the tubulin proteins whose malfunction in Alzheimer's disease affects neuronal transport and stability. "TUBA1B" and "TUBB3" are largely overlooked by all other tested measures, except for Katz centrality, which ranks them as the second and third most important nodes in the network.

Similarly to what we observe for the case of the `hsa05022` pathway, the "Aβ40" node is highlighted as the most important node by all measures except *stsc*. In fact, *stsc* assigns "Aβ40" the sixth position in the rankings.

The rankings are reported in Table 4.8.

hsa04010

In this example, $\mathbf{d}^{(in)}+\mathbf{d}^{(out)}$ and *stsc* agree on the three most important nodes in the network: "MAPK8", "MAPK1", "MAPK14", which are, respectively, JNK, ERK, and p38 kinases responsible for proliferation and differentiation under physiological or stress conditions.

In general, we observe that the rankings given by *stsc* show some similarities with those given by $\mathbf{d}^{(in)}+\mathbf{d}^{(out)}$, $\boldsymbol{\pi}+\boldsymbol{\pi}^{(rev)}$, and $\mathbf{h}+\mathbf{a}$. The rankings given by $\mathbf{k}^{(b)} + \mathbf{k}^{(r)}$, on the other hand, highlights nodes that are significantly different from the one highlighted by other measures. Remarkably, *stsc* lists among the ten most important nodes "FOS" and "ATF4" , which are ranked as the first and second nodes by $\mathbf{k}^{(b)} + \mathbf{k}^{(r)}$. This is somewhat expected, as both "FOS" and "ATF4" are transcription factors that act as final effectors of the pathway involving "MAPK8", "MAPK1", "MAPK14".

The rankings are reported in Table 4.9.

Table 4.9: Top 10 nodes of the `hsa04010` network ranked using *stsc* and the other centrality measures described in Section 4.4

$\mathbf{d}^{(in)} + \mathbf{d}^{(out)}$	$\boldsymbol{\pi} + \boldsymbol{\pi}^{(rev)}$	$\mathbf{k}^{(b)} + \mathbf{k}^{(r)}$	$\mathbf{h} + \mathbf{a}$	*stsc*
MAPK8	MAPK14	FOS	MAPK8	MAPK8
MAPK1	MAPK8	ATF4	MAP2K7	MAPK1
MAPK14	MAPK1	ELK1	MAP3K11	MAPK14
RRAS2	RRAS2	ELK4	PPM1A	RRAS2
MAP3K7	MAP3K7	MAX	MAP2K3	FOS
MAP2K7	MAP2K3	MAPK8IP2	MAP2K6	MAP3K7
MAP2K3	MAP2K6	MAPK1	MAP3K5	MAP2K3
MAP2K6	TRADD	TRADD	MAPK8IP2	MAP2K6
MAP3K5	TNFRSF1A	CACNG3	DUSP3	ATF4
TRAF2	TAB1	TP53	TRAF2	TAB1

4.6. Conclusions

We proposed a new centrality measure for DAGs based on the fraction of source-to-sink walks that visit a node. This new score, which we called *source-to-sink communicability* (or *stsc* for short), has among its most important features its non-locality, which is the ability to give high scores to nodes with relatively low degrees but effective bridging capabilities. Furthermore, as expected, we experimented also that *stsc* can establish hierarchies between different "bridge"nodes by providing informative scores that summarise their importance in source-to-sink communications. We included also an efficient strategy for computing our score and updating it after the removal of an edge from the graph. Experiments on synthetic DAGs show that *stsc* can be applied effectively even to large DAGs. They also show that, in the case of topologically sorted DAGs, i.e., networks whose adjacency matrix is upper triangular, the update formula significantly reduces the computational cost of computing the underlying centrality index from scratch.

In this chapter we focused on protein interaction networks, in which counting the number of pathways can be used to identify nodes whose failure would likely cause severe disruption of the underlying biological processes. However, the proposed score can be applied in many other practical applications outside the biological context, e.g., in articles citation networks [39], which are typically directed and acyclic.

5. Conclusions and future work

The primary objective of this thesis is to propose methods to approximate walk-based measures after a network perturbation, providing an efficient alternative to their full recomputation which can be prohibitive in the case of large or frequently changing networks. This topic was addressed in Chapters 2 and 3. In Chapter 2 we derived two "walk-informed" algorithms to update Katz centrality after the removal of nodes or edges from a simple graph. In Chapter 3 we established upper and lower bounds on the changes in f-communicability and related network indices caused by perturbing (adding, deleting, or modifying the weights of) a set of edges in a graph.

The second objective was addressed in Chapter 4, where we introduced a novel walk-based centrality measure for directed acyclic networks that is computationally efficient, updatable after edge removal, and well-suited to the analysis of some protein interaction networks.

Central to this thesis was the derivation of several combinatorial results concerning the enumeration of walks passing through a specific set of nodes or edges in a (simple, directed, or directed and acyclic) graph. Another contribution was the introduction of a special class of analytic functions that includes Mittag-Leffler functions, and whose properties were crucial to proving our bounds on f-communicability in Chapter 3.

Several potential directions could be considered for future research. For example, it would be interesting to generalise the methods from Chapter 2 to non-backtracking [57] and/or time evolving [11, 58] Katz centrality. This appears to be a challenging task, as it would require proving results for counting all non-backtracking walks that visit a set or nodes or edges in a (possibly time-evolving) graph. Preliminary findings (not included in this thesis) suggest that this is a promising direction for such an extension.

Similar observations hold for the results presented in Chapter 3, with the additional difficulties arising from the presence of weighted graphs (see [12]). Despite these challenges, extending our bounds to the non-backtracking/time-evolving case could be highly beneficial, as it would reduce the computational

burden of recomputing these measures, which can be an issue in the case of large networks.

Regarding the new centrality measure introduced in Chapter 4, it would be worthwhile to explore methods for approximately computing and updating it. These methods would be valuable in the context of large-scale DAGs that are not necessarily topologically sorted. Moreover, a more in-depth analysis of its application to biological pathways would be beneficial for biologists wishing to identify crucial nodes in such protein interaction networks.

Bibliography

[1] A. H. Al-Mohy and N. J. Higham. Computing the action of the matrix exponential, with an application to exponential integrators. *SIAM Journal on Scientific Computing*, 33(2):488–511, 2011.

[2] R. Albert, I. Albert, and G. L. Nakarado. Structural vulnerability of the north american power grid. *Phys. Rev. E*, 69:025103, Feb 2004.

[3] U. Alon. Network motifs: theory and experimental approaches. *Nature Reviews Genetics*, 8:450–461, 2007.

[4] G. Alterovitz, R. Benson, and M. Ramoni. *Automation in Proteomics and Genomics: An Engineering Case-Based Approach.* John Wiley & Sons, Chichester, West Sussex, UK, 03 2009.

[5] F. Arrigo and M. Benzi. Edge modification criteria for enhancing the communicability of digraphs. *SIAM Journal on Matrix Analysis and Applications*, 37(1):443–468, 2016.

[6] F. Arrigo and M. Benzi. Updating and downdating techniques for optimizing network communicability. *SIAM Journal on Scientific Computing*, 38(1):B25–B49, 2016.

[7] F. Arrigo, D. Bertaccini, and A. Filippo. Updating Katz centrality by counting walks. *SIAM Journal on Matrix Analysis and Applications*, 46(4):2449–2474, 2025.

[8] F. Arrigo and F. Durastante. Mittag-Leffler Functions and their Applications in Network Science. *SIAM Journal on Matrix Analysis and Applications*, 42(4):1581–1601, 2021.

[9] F. Arrigo and A. Filippo. Estimating f-communicability in digraphs after edge perturbation. In preparation.

[10] F. Arrigo, D. J. Higham, and V. Noferini. A theory for backtrack-downweighted walks. *SIAM Journal on Matrix Analysis and Applications*, 42(3):1229–1247, 2021.

[11] F. Arrigo, D. J. Higham, V. Noferini, and R. Wood. Dynamic Katz and related network measures. *Linear Algebra and its Applications*, 655:159–185, 2022.

[12] F. Arrigo, D. J. Higham, V. Noferini, and R. Wood. Weighted enumeration of nonbacktracking walks on weighted graphs. *SIAM Journal on Matrix Analysis and Applications*, 45(1):397–418, 2024.

[13] M. Ashtiani, A. Salehzadeh-Yazdi, Z. Razaghi-Moghadam, H. Hennig, O. Wolkenhauer, M. Mirzaie, and M. Jafari. A systematic survey of centrality measures for protein-protein interaction networks. *BMC Systems Biology*, 12(1):80, 2018.

[14] A.-L. Barabási and Z. N. Oltvai. Network biology: understanding the cell's functional organization. *Nature reviews. Genetics*, 5(2):101—113, February 2004.

[15] A.-L. Barabási and M. Pósfai. *Network Science*. Cambridge University Press, Cambridge, 2016.

[16] M. Bastian, S. Heymann, and M. Jacomy. Gephi: An open source software for exploring and manipulating networks. *Proceedings of the International AAAI Conference on Web and Social Media*, 3(1):361–362, Mar. 2009.

[17] B. Beckermann, D. Kressner, and M. Schweitzer. Low-Rank Updates of Matrix Functions. *SIAM Journal on Matrix Analysis and Applications*, 39(1):539–565, 2018.

[18] M. Benzi, D. Bertaccini, F. Durastante, and I. Simunec. Non-local network dynamics via fractional graph laplacians. *Journal of Complex Networks*, 8(3):cnaa017, 08 2020.

[19] M. Benzi and P. Boito. Matrix functions in network analysis. *GAMM-Mitteilungen*, 43(3):e202000012, 2020.

[20] M. Benzi, E. Estrada, and C. Klymko. Ranking hubs and authorities using matrix functions. *Linear Algebra and its Applications*, 438(5):2447–2474, 2013.

[21] M. Benzi and C. Klymko. Total communicability as a centrality measure. *Journal of Complex Networks*, 1(2):124–149, 05 2013.

[22] M. Benzi and C. Klymko. On the limiting behavior of parameter-dependent network centrality measures. *SIAM Journal on Matrix Analysis and Applications*, 36(2):686–706, 2015.

[23] D. S. Bernstein and C. F. Van Loan. Rational matrix functions and rank-1 updates. *SIAM Journal on Matrix Analysis and Applications*, 22(1):145–154, 2000.

[24] D. Bertaccini, L. Chiricosta, and A. Filippo. Source-to-sink communicability: a new centrality measure for directed acyclic networks. *Numerical Algorithms*, 2025.

[25] 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.

[26] D. Bertaccini and F. Durastante. Nonlocal diffusion of variable order on complex networks. *International Journal of Computer Mathematics: Computer Systems Theory*, 7(3):172–191, 2022.

[27] D. Bertaccini and A. Filippo. A proposal for ranking through selective computation of centrality measures. *PLOS ONE*, 18(9):1–16, 09 2023.

[28] P. Bonacich. Factoring and weighting approaches to status scores and clique identification. *Journal of mathematical sociology*, 2(1):113–120, 1972.

[29] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1):107–117, 1998.

[30] D. Cartwright, F. Harary, and R. Z. Norman. *Structural Models: An Introduction to the Theory of Directed Graphs*. John Wiley & Sons, New York, NY, 1965.

[31] 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*, pages 325–333, United States, 2014. SIAM.

[32] S. Cipolla, F. Durastante, and B. Meini. Enforcing katz and pagerank centrality measures in complex networks. *SIAM Journal on Mathematics of Data Science*, 7(3):1514–1539, 2025.

[33] S. Cipolla, F. Durastante, and F. Tudisco. Nonlocal PageRank. *ESAIM: M2AN*, 55(1):77–97, 2021.

[34] R. Cohen, K. Erez, D. ben Avraham, and S. Havlin. Breakdown of the internet under intentional attack. *Phys. Rev. Lett.*, 86:3682–3685, Apr 2001.

[35] K. Das, S. Samanta, and M. Pal. Study on centrality measures in social networks: a survey. *Social Network Analysis and Mining*, 8(1):13, 2018.

[36] T. Davis, W. Hager, and I. Duff. SuiteSparse. *http://faculty.cse.tamu.edu/davis/suite sparse.html*, 2014.

[37] T. A. Davis. *Direct Methods for Sparse Linear Systems*. SIAM, 2006.

[38] O. De la Cruz Cabrera, J. Jin, S. Noschese, and L. Reichel. Communication in complex networks. *Applied Numerical Mathematics*, 172:186–205, 2022.

[39] D. J. de Solla Price. Networks of scientific papers. *Science*, 149(3683):510–515, 1965.

[40] E. Dudkina, M. Bin, J. Breen, E. Crisostomi, P. Ferraro, S. Kirkland, J. Mareček, R. Murray-Smith, T. Parisini, L. Stone, S. Yilmaz, and R. Shorten. A comparison of centrality measures and their role in controlling the spread in epidemic networks. *International Journal of Control*, 97(6):1325–1340, 2024.

[41] J. A. Dunne, R. J. Williams, and N. D. Martinez. Food-web structure and network theory: The role of connectance and size. *Proceedings of the National Academy of Sciences*, 99(20):12917–12922, 2002.

[42] 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(4):558–567, 2002.

[43] P. Erdös and A. Rényi. On random graphs I, 1959.

[44] E. Estrada. *The Structure of Complex Networks: Theory and Applications*. Oxford University Press, Oxford, UK, 1st edition, 2011.

[45] E. Estrada and N. Hatano. Communicability in complex networks. *Phys. Rev. E*, 77:036111, Mar 2008.

[46] E. Estrada, N. Hatano, and M. Benzi. The physics of communicability in complex networks. *Physics Reports*, 514(3):89–119, 2012. The Physics of Communicability in Complex Networks.

[47] E. Estrada and D. J. Higham. Network properties revealed through matrix functions. *SIAM review*, 52(4):696–714, 2010.

[48] E. Estrada, D. J. Higham, and N. Hatano. Communicability betweenness in complex networks. *Physica A: Statistical Mechanics and its Applications*, 388(5):764–774, 2009.

[49] E. Estrada and J. A. Rodríguez-Velázquez. Subgraph centrality in complex networks. *Phys. Rev. E*, 71:056103, May 2005.

[50] R. Fagin, R. Kumar, and D. Sivakumar. Comparing top k lists. *SIAM J. Discret. Math.*, 17:134–160, 2003.

[51] K. Foster, S. Muth, J. Potterat, and R. Rothenberg. A faster Katz status score algorithm. *Computational & Mathematical Organization Theory*, 7:275–285, 12 2001.

[52] L. C. Freeman. Centrality in social networks conceptual clarification. *Social Networks*, 1(3):215–239, 1978.

[53] M. Ghasemi, H. Seidkhani, and F. Tamimi. Centrality measures in biological networks. *Current Bioinformatics*, 9, 05 2014.

[54] D. F. Gleich. Pagerank beyond the web. *SIAM Review*, 57(3):321–363, 2015.

[55] G. H. Golub and G. Meurant. *Matrices, Moments and Quadrature with Applications*. Princeton University Press, 2010.

[56] G. H. Golub and C. F. Van Loan. *Matrix computations*. John Hopkins University Press, Baltimore, MD, 3rd edition, 1996.

[57] 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(1):310–341, 2018.

[58] P. Grindrod, M. C. Parsons, D. J. Higham, and E. Estrada. Communicability across evolving networks. *Phys. Rev. E*, 83:046120, Apr 2011.

[59] R. Guimerà, S. Mossa, A. Turtschi, and L. Amaral. The worldwide air transportation network: Anomalous centrality, community structure, and cities' global roles. *Proceedings of the National Academy of Sciences of the United States of America*, 102:7794–9, 06 2005.

[60] S. Güttel. Rational krylov approximation of matrix functions: Numerical methods and optimal pole selection. *GAMM-Mitteilungen*, 36(1):8–31, 2013.

[61] J. Hadamard. Théorème sur les séries entières. *Acta Mathematica*, 22(none):55 – 63, 1900.

[62] M. W. Hahn and A. D. Kern. Comparative genomics of centrality and essentiality in three eukaryotic protein-interaction networks. *Molecular Biology and Evolution*, 22(4):803–806, 12 2004.

[63] F. Harary. *Graph Theory*. Westview Press, 1969.

[64] D. J. Higham and A. Taylor. CONTEST: A controllable test matrix toolbox for MATLAB. *ACM Trans. Math. Softw.*, 35(4), feb 2009.

[65] N. Higham. *Functions of Matrices: Theory and Computation*. SIAM, 2008.

[66] P. Holme, B. J. Kim, C. N. Yoon, and S. K. Han. Attack vulnerability of complex networks. *Phys. Rev. E*, 65:056109, May 2002.

[67] R. Horn and C. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, 1994.

[68] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge university press, 2nd edition, 2012.

[69] S. Iyer, T. Killingback, B. Sundaram, and Z. Wang. Attack robustness and centrality of complex networks. *PLOS ONE*, 8(4):1–17, 04 2013.

[70] B. Jackson and A. D. Sokal. Maxmaxflow and counting subgraphs. *Electron. J. Comb.*, 17, 2007.

[71] H. Jeong, S. Mason, A.-L. Barabasi, and Z. Oltvai. Lethality and centrality in protein networks. *Nature*, 411:41–2, 06 2001.

[72] M. Kanehisa and S. Goto. KEGG: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Research*, 28(1):27–30, 2000.

[73] L. Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18:39–43, 1953.

[74] M. Kendall. *Rank Correlation Methods*. Griffin, Oxford, UK, 1948.

[75] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, Sept. 1999.

[76] J. M. Kleinberg, R. Kumar, P. Raghavan, S. Rajagopalan, and A. S. Tomkins. The web as a graph: Measurements, models, and methods. In T. Asano, H. Imai, D. T. Lee, S.-i. Nakano, and T. Tokuyama, editors, *Computing and Combinatorics*, pages 1–17, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.

[77] D. Koschützki and F. Schreiber. Centrality analysis methods for biological networks and their application to gene regulatory networks. *Gene Regulation and Systems Biology*, 2:193 – 201, 2008.

[78] T. B. Köse, J. Li, and A. Ritz. Growing directed acyclic graphs: Optimization functions for pathway reconstruction algorithms. *Journal of Computational Biology*, 30(7):814–828, 2023.

[79] W. Li, M. Chu, and J. Qiao. A pruning feedforward small-world neural network based on Katz centrality for nonlinear system modeling. *Neural Networks*, 130:269–285, 2020.

[80] S. Massei and F. Tudisco. Optimizing network robustness via Krylov subspaces. *ESAIM: M2AN*, 58(1):131–155, 2024.

[81] C. Mitrea, Z. Taghavi, B. Bokanizad, S. Hanoudi, R. Tagett, M. Donato, C. Voichiţa, and S. Draghici. Methods and approaches in the topology-based analysis of biological pathways. *Frontiers in Physiology*, 4:278, 10 2013.

[82] G. M. Mittag-Leffler. Sur la nouvelle fonction $e_\alpha(x)$. *Comptes Rendus de l'Académie des Sciences de Paris*, 137:554–558, 1903.

[83] P. Naderi Yeganeh, C. Richardson, E. Saule, A. Loraine, and M. Mostafavi. Revisiting the use of graph centrality models in biological pathway analysis. *BioData Mining*, 13, 06 2020.

[84] E. Nathan and D. Bader. Incrementally updating Katz centrality in dynamic graphs. *Social Network Analysis and Mining*, 8:26, 03 2018.

[85] E. Nathan and D. A. Bader. Approximating personalized Katz centrality in dynamic graphs. In *Parallel Processing and Applied Mathematics*, pages 290–302, Cham, 2018. Springer.

[86] M. E. J. Newman. The Structure and Function of Complex Networks. *SIAM Review*, 45(2):167–256, 2003.

[87] M. E. J. Newman. *Networks: An Introduction*. Oxford University Press, Oxford, UK, 1st edition, 2010.

[88] V. Noferini, S. Vrontos, and R. Wood. Efficient computation of f-centralities and nonbacktracking centrality for temporal networks. arXiv.2410.17352.

[89] V. Noferini and R. Wood. Efficient computation of Katz centrality for very dense networks via negative parameter Katz. *Journal of Complex Networks*, 12(5):cnae036, 2024.

[90] F. Olver, D. Lozier, R. Boisvert, and C. Clark. *The NIST Handbook of Mathematical Functions*. Cambridge University Press, New York, NY, 2010.

[91] 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(4):1521–1546, 2018.

[92] Y. Saad. Analysis of some krylov subspace approximations to the matrix exponential operator. *SIAM Journal on Numerical Analysis*, 29(1):209–228, 1992.

[93] Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, 2003.

[94] D. Salwinski. The continuous binomial coefficient: An elementary approach. *The American Mathematical Monthly*, 125(3):231–244, 2018.

[95] M. Schweitzer. Sensitivity of matrix function based network communicability measures: computational methods and a priori bounds. *SIAM Journal on Matrix Analysis and Applications*, 44(3):1321–1348, 2023.

[96] 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(1):124–127, 1950.

[97] B. Snel, G. Lehmann, P. Bork, and M. A. Huynen. String: a web-server to retrieve and display the repeatedly occurring neighbourhood of a gene. *Nucleic Acids Research*, 28(18):3442–3444, 09 2000.

[98] D. Tenenbaum and B. Maintainer. Keggrest: Client-side rest access to the kyoto encyclopedia of genes and genomes (KEGG). R package, version 1.46.0, 2024.

[99] P. Van Mieghem. *Graph Spectra for Complex Networks*. Cambridge University Press, 2011.

[100] R. Varga. *Matrix Iterative Analysis*. Springer Series in Computational Mathematics. Springer Berlin Heidelberg, 2009.

[101] S. Wasserman and K. Faust. Social network analysis: Methods and applications, 1994.

[102] 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(9):e24306, 2011.