Complex network analysis using modulus of families of walks

by

Heman Shakeri

B.S., Amirkabir University (Tehran Polytechnic), 2008

M.S., Amirkabir University (Tehran Polytechnic), 2011

AN ABSTRACT OF A DISSERTATION

submitted in partial fulfillment of the
requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Electrical and Computer Engineering
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2017

Abstract

The modulus of a family of walks quantifies the richness of the family by favoring having many short walks over a few longer ones. In this dissertation, we investigate various families of walks to study new measures for quantifying network properties using modulus. The proposed new measures are compared to other known quantities. Our proposed method is based on walks on a network, and therefore will work in great generality. For instance, the networks we consider can be directed, multi-edged, weighted, and even contain disconnected parts.

We study the popular centrality measure known in some circles as information centrality, also known as effective conductance centrality. After reinterpreting this measure in terms of modulus of families of walks, we introduce a modification called shell modulus centrality, that relies on the egocentric structure of the graph. Ego networks are networks formed around egos with a specific order of neighborhoods. We then propose efficient analytical and approximate methods for computing these measures on both directed and undirected networks. Finally, we describe a simple method inspired by shell modulus centrality, called *general degree*, which improves simple degree centrality and could prove to be a useful tool for practitioners in the applied sciences. General degree is useful for detecting the best set of nodes for immunization.

We also study the structure of loops in networks using the notion of modulus of loop families. We introduce a new measure of network clustering by quantifying the richness of families of (simple) loops. Modulus tries to minimize the expected overlap among loops by spreading the expected link-usage optimally. We propose weighting networks using these expected link-usages to improve classical community detection algorithms. We show that the proposed method enhances the performance of certain algorithms, such as spectral partitioning and modularity maximization heuristics, on standard benchmarks.

Computing loop modulus benefits from efficient algorithms for finding shortest loops, thus we propose a deterministic combinatorial algorithm that finds a shortest cycle in graphs. The proposed algorithm reduces the worst case time complexity of the existing combinatorial algorithms to $\mathcal{O}(nm)$ or $\mathcal{O}(\langle k\rangle n^2\log n)$ while visiting at most $m-n+1$ cycles (size of cycle basis). For most empirical networks with average degree in $\mathcal{O}(n^{1-\epsilon})$ our algorithm is subcubic.

Complex network analysis using modulus of families of walks

by

Heman Shakeri

B.S., Amirkabir University (Tehran Polytechnic), 2008

M.S., Amirkabir University (Tehran Polytechnic), 2011

A DISSERTATION

submitted in partial fulfillment of the
requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Electrical and Computer Engineering
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2017

Approved by:

Co-Major Professor
Caterina Scoglio

Approved by:

Co-Major Professor
Pietro Poggi-Corradini

Copyright

Abstract

The modulus of a family of walks quantifies the richness of the family by favoring having many short walks over a few longer ones. In this dissertation, we investigate various families of walks to study new measures for quantifying network properties using modulus. The proposed new measures are compared to other known quantities. Our proposed method is based on walks on a network, and therefore will work in great generality. For instance, the networks we consider can be directed, multi-edged, weighted, and even contain disconnected parts.

We study the popular centrality measure known in some circles as information centrality, also known as effective conductance centrality. After reinterpreting this measure in terms of modulus of families of walks, we introduce a modification called shell modulus centrality, that relies on the egocentric structure of the graph. Ego networks are networks formed around egos with a specific order of neighborhoods. We then propose efficient analytical and approximate methods for computing these measures on both directed and undirected networks. Finally, we describe a simple method inspired by shell modulus centrality, called *general degree*, which improves simple degree centrality and could prove to be a useful tool for practitioners in the applied sciences. General degree is useful for detecting the best set of nodes for immunization.

We also study the structure of loops in networks using the notion of modulus of loop families. We introduce a new measure of network clustering by quantifying the richness of families of (simple) loops. Modulus tries to minimize the expected overlap among loops by spreading the expected link-usage optimally. We propose weighting networks using these expected link-usages to improve classical community detection algorithms. We show that the proposed method enhances the performance of certain algorithms, such as spectral partitioning and modularity maximization heuristics, on standard benchmarks.

Computing loop modulus benefits from efficient algorithms for finding shortest loops, thus we propose a deterministic combinatorial algorithm that finds a shortest cycle in graphs. The proposed algorithm reduces the worst case time complexity of the existing combinatorial algorithms to $\mathcal{O}(nm)$ or $\mathcal{O}(\langle k\rangle n^2 \log n)$ while visiting at most $m - n + 1$ cycles (size of cycle basis). For most empirical networks with average degree in $\mathcal{O}(n^{1-\epsilon})$ our algorithm is subcubic.

Table of Contents

List of Figures

List of Tables

Acknowledgments

I am grateful to my advisors Dr. Scoglio and Dr. Poggi-Corradini for their support, patience and encouragement during these four years. They have guided me in research and inspired me to contribute, and enjoy my years as a PhD student. A very special gratitude goes to Dr. Albin; he was a great pleasure to work with and I learnt a lot from his countless priceless advises.

I want to express my appreciation to Dr. Prakash for his valuable time and advises during my PhD examinations. I am also grateful to Dr. Easton for donating his time to be the external chair of my defense and for advancing my knowledge with his wonderful course “Network Flow”.

Special thanks to my friends in Network Science and Engineering Lab and NODE: Haotian Wu, Xin Li, Aram Vajdi, Futing Fan, Faryad Sahneh, and all others. I would like to thank Dr. Michael Higgins and his research group for their precious feedbacks on my works. Many thanks to my friends in K-State, Nazif, Hazhar, Sina, Peyman, Sajed, Yeng, and Adam.

I would like to thank my parents and my family Nooshin, Mansour, Peyman, Hiwa, Shahram, Behnam, and Ali, for their spiritual support through my graduate studies.

There are not enough words to describe how grateful I am to my best friend in life, my wife Behnaz Moradi, who supports me with his unconditional love through all my days.

Dedication

To Behnaz

Preface

This dissertation with title "Complex Networks Analysis Using Modulus of Families of Walks" is submitted for the degree of Doctor of Philosophy in the Department of Electrical and Computer Engineering at Kansas State University. The research has been performed under the supervision of Prof. Caterina Scoglio and Pietro Poggi-Corradini. Most of the work is comprised from the following set of published or submitted peer-reviewed journals (with the corresponding chapter in the dissertation):

1. **H. Shakeri**, P Poggi-Corradini, C. Scoglio, and N. Albin. "Modulus of families of loops with applications in network analysis". *Physical review E 95, 012316.* (2017). (Chapter 5)

2. **H. Shakeri**, N. Albin, P. Poggi-Corradini, and C. Scoglio. "Generalized Network Measures Based on Modulus of Families of Walks", *Journal of Computational and Applied Mathematics 307, 307-318* (2016). (Chapter 3)

3. **H. Shakeri**, B. Moradi, P. Poggi-Corradini, N. Albin, and C. Scoglio, "Egocentric Centrality Measures and General Degree". *Submitted for publication.* (Chapter 4)

4. **H. Shakeri,** N. Albin, F. D. Sahneh, P. Poggi-Corradini, and C. Scoglio. "Maximizing Algebraic Connectivity in Interconnected Networks". *Physical review E 93, 030301(R)* (2016). (Appendix A)

5. **H. Shakeri**, F. D. Sahneh, C. Scoglio, P. Poggi-Corradini, and V. M. Preciado. "Optimal information dissemination strategy to promote preventive behaviors in multilayer epidemic networks". *Mathematical biosciences and engineering: MBE* 12 (3), 609-623 (2015). (Appendix B)

6. F. D. Sahneh, A. Vajdi, **H. Shakeri**, F. Futing, and C. Scoglio. "GEMF: Generalized epidemic modelling framework software (GEMF)". *arXiv preprint arXiv:1604.02175* (2016). (Appendix C)

7. **H. Shakeri**, B. Moradi, P. Poggi-Corradini, N. Albin, and C. Scoglio. "Finding minimum weight cycle in graphs using Dijkstra's algorithm". *Submitted for publication.* (Chapter 5)

This research is focused on using the notion of modulus of families of walks in network analysis. Therefore, articles with less focus on modulus concepts are reported in the appendix.

This research was partially funded by National Science Foundation grants DMS-1201427 and DMS-1515810 and assistantships from the Electrical and Computer Engineering Department at Kansas State University and the Department of Mathematics at Kansas State University.

The experiments, and simulations were performed by Heman Shakeri and will be open for further criticisms and questions.

Chapter 1

Introduction

1.1 Introduction

Network analysis has become a way to understand complex interconnected systems in different disciplines, such as social sciences, engineering, statistics, biological systems, etc. Universal properties of networks in different domains have been discovered. However, the rapid growth of this field in the recent two decades also brought challenges such as scalability and domain-specific needs. This leads to the introduction of ad-hoc methods that lack theoretical background. As an example, several methods that aim at ranking the nodes based on their spreading abilities, have been proposed[21–23]. Moreover, there are studies that focus on the applicability of existing methods for specific applications[24], claiming that "network analysis has been driven much more strongly by its methods than by its theories." Furthermore, a method that is designed for a specific domain can be ineffective for another.

Most of the classical measures are based on simplified concepts such as geodesic distances or counting simple structures, e.g. triangles for clustering coefficients. More sophisticated methods, such as current flow closeness centrality[7], or random walk betweenness centrality[25], lack scalability and also, need special accommodations such as "symmetric" graphs (undirected), or full ranked matrices (connectedness). As an example, in a piece of the Facebook network in Figure 1.1, the clustering coefficient defined based on average number of

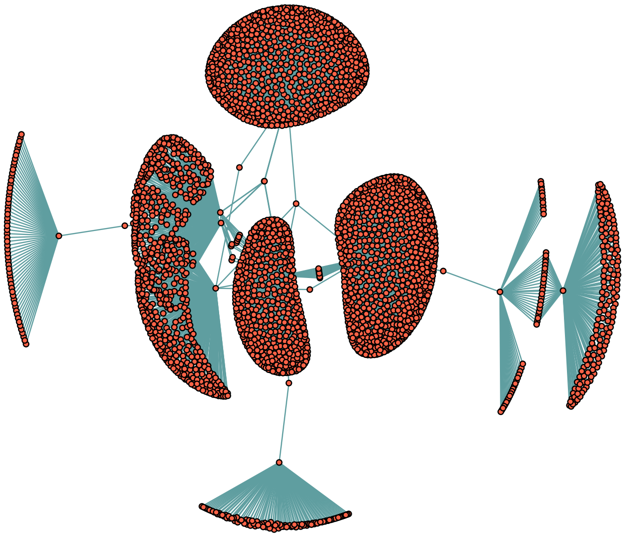


Figure 1.1: An excerpt of Facebook network with $n = 2888$ and $m = 2981$. Edges represent friendships between nodes[3] with clustering coefficient 0.03%.

existing triangles over the number of possible triangles near a given node, cannot quantify the structure of the loops in the network. A trade-off is required to be able to analyze the local and global structure of the network, and it is not clear how to weigh these different aspects. Our modulus technique will provide a synthesis between local and global properties.

In some cases, researchers have discovered important aspects of a network by analyzing the overall network topology. One example is the number of short cycles in a graph. Methods have appeared that take this into account by counting triangles, squares, etc, and then use this information to find communities in a network. For instance, Radicchi et al. count the number of short loops that pass through a given link as a local measure for clustering[26]. To extend the method in[26] for low clustered networks, Vragovic *et al.* in[27] consider general loops (with any length) passing through a node. However, according to[28] its results are not satisfying compared to standard clustering methods. The authors in[29] define a new weighting for a network to improve modularity maximization methods for finding communities with sizes smaller than the resolution limit[30]. The weighting for a link comes from how many loops with length 3 and 4 it forms with the adjacent links. They show the effectiveness of their method on Lancichinetti, Fortunato, and Radicchi (LFR) benchmark networks. Also the authors in[31] propose weighting a network with a combination of link-betweenness centrality[32] and another measure called *common neighbor ratio* to enhance community identification. These are ad-hoc methods that lack theoretical justification. Modulus can offer a unified framework to these issues with a common theoretical background.

The requirement for complete knowledge of the network (sociocentric data) is another drawback of existing methods. In the social sciences, anonymizing the data to protect the privacy of network entities and also ethical reasons can prevent scientists from accessing the complete network. Egonetworks or also known as neighborhood networks and are considered as samples of the underlying network. They are constructed around focal nodes (egos) and allow for more flexible data collection and inexpensive computational costs. Thus egonetworks are increasingly popular among social scientists[33–36].

Several attempts have been made to address the need of egocentric measures[21;37;38]. In addition to experimental studies, more theoretical justification is required. Marsden[39] proposed the egocentric versions of classical sociocentric centrality and betweenness centrality measures introduced in Freeman's seminal works[32;40]. He considers the first order neighborhood of ego and shows that classical centrality measures reduce to degree centrality. However, the egocentric betweenness centrality can have different biases. Egocentric measures show more stability[41] against network sampling and less sensitivity to measurement errors[42]. In Chapters 3 and 4, we study scalable centrality measures for ego networks based on modulus.

Real networks contain closely connected subnetworks with local structural patterns characterized by their richness of loop[43]. Loops offer a richer set of pathways compared to treelike topologies; thus rich loop structures improve network robustness[44] and impact propagating and transporting processes in networks[45]. Previous approaches on the analysis of loop structures focused on loops with lengths of order 3–5 separately[46;47] and few such as[48;49] emphasized the role of higher order loops to characterize their overall structures. We consider assessing the loop structure of a network using modulus. Thus we will take into account loops of any length as well as their relative position. This allows us to analyze network transitivity measures such as clustering coefficients, and to provide more information for community detection algorithms.

In conclusion, the abstract framework of modulus allows us to analyze a given network using a variety of families of walks, that can be defined in response to specific needs. We develop new measures based on the richness of these families of walks that generalize existing classical measures, while remaining scalable to address practical applications. We consider different scenarios such as lack of complete network data or networks with added constraints, such as directedness. Our focus for applications is investigating networked epidemic processes and developing tools to mitigate disease outbreaks in contact networks or promote awareness in information dissemination networks.

This dissertation studies applications of the modulus of families of walks on networks developed in[50–52]. This is a discrete analog of the classical theory of modulus of curve families in complex analysis[53]. Although modulus on networks has been studied under several different guises, see[54–57], it is not as well understood as in the continuum setting.

Modulus is a way of measuring the richness of certain families of objects on a network, such as loops, walks, trees, etc, and is a discrete analog of the classical theory of modulus of curve families in complex analysis[53]. Although modulus on networks is not a new concept (see[55;56]), it is not as well developed as in the continuum setting. Our study of modulus of walks on networks originated from[58] in which the authors compared it to a new geometric measure they called "epidemic quasimetric". In[50], the authors showed that modulus is a standard convex optimization problem. Continuity and smoothness properties of modulus on networks were considered in[51]. A probabilistic interpretation provided in[52].

Modulus is a versatile tool to analyze networks. Different types of families of walks can be used to learn about different aspects of the network. In[59], we introduced centrality measures based on various families of walks that can be computed on directed or undirected, weighted or unweighted, and even disconnected networks. These measures do not necessarily have to consider the whole network. We applied them to detect influential sections of the network, ranking the nodes, and we explored applications to improve vaccination strategies for reducing the risk of epidemics. The applications to epidemic spreading were further studied in[60], where the authors used modulus to analyze the concept of Epidemic Hitting Time.

We explore the versatility of modulus of families of walks, demonstrating that it provides a powerful approach to the study of networks. We describe different problems that can be handled by various classes of families of walks. Furthermore, we propose measures based on these families that can be applied in a general framework, handling directed or undirected, weighted or unweighted, and disconnected networks, while the amount of information extracted from a network can be adjusted with high accuracy.

1.2 Broader impact

Better measures to characterize and analyze networks are crucial for generating accurate models and predicting the outcome of networked processes.

We investigate the concept of modulus of family of walks on networks that gives a method for quantifying the richness of walks. Modulus generalizes concepts of connectivity ranging from shortest path and minimum cut to effective conductance. Thus modulus helps understanding the network functions, such as synchronization, network topology characteristics, such as clusters, and network properties, such as robustness. Therefore, applications of our methods are wide-ranging and span from engineering to biology and data science.

A particular application of this theory relates to modeling and simulation of epidemics, such as Ebola and flu. We suggest effective mitigation strategies by identifying best sets of nodes to immunize.

Analyzing the topology of the network with modulus is crucial from unsupervised learning to infer the relationship between different entities in system biology. Moreover, combining results from modulus with operations theory problems is another field of application.

1.3 Contributions

Our contributions can be summarized as the following. We

1. Introduce modulus of families of walks as a comprehensive method for analyzing network structure

2. Develop a flexible network analysis paradigm that can be adapted based on user interest. For example, same concepts can be applied for families of spanning trees or families of matchings.

3. Apply the proposed measures to problems such as detecting influential parts of networks that can serve as major spreaders.

4. Rank most influential nodes in networks by analyzing walks as the generic pathways of influence.

5. Develop efficient vaccination methods to mitigate the spread of infection diseases in networks.

6. Introduce a powerful egocentric network measure called shell modulus.

7. Define general degree, as a simple centrality measure that enhances degree centrality.

8. Introduce a generic approach to analyze loops structures in the network that consider local loop topologies with an eye on the entire network.

9. Quantify richness of loops and introduce a clustering measure based on modulus of families of loop.

10. Find the probability of usage for each link in important loops and use it as a measure of affinity between nodes to enhance network partitioning.

11. Develop an improved combinatorial algorithm for finding shortest cycle in weighted graphs.

1.4 Organization

The fundamental concepts of modulus of families of walks are presented in Chapter 2. We introduce shell modulus in Chapter 3 and propose a powerful framework to design centrality measures. Ego networks are discussed in Chapter 4 with ways to measure ego's centralities. Modulus of family of loops is discussed in Chapter 5, with efficient algorithm to find the shortest cycle in weighted networks. Closing thoughts with future direction of this research are in Chapter 2.

Epidemic simulations are done with GEMFPy a stochastic simulator for epidemic processes on networks developed by the author. We briefly describe it in Appendix C. During this work, a lot of other ideas brewed and flourished, we will discuss two of them in Appendices A and B.

Chapter 2

Modulus of Families of Walks on Networks

In this chapter, we introduce our notations and definitions. We offer a brief review of modulus of family of simple walks here, together with basic algorithms for its computation. To delve deeper, we encourage interested reader to see[50–52;61]. Starting from basic definitions, let $G = (V, E)$ be a network with node set V and link set E. The cardinalities of V and E are denoted by n and m respectively. Let $p \geq 1$ and let $w : E \to (0, \infty)$ be a positive weight function representing a generalized edge conductivity (for undirected networks weights are binary values).

2.1 Families of walks

A *walk* γ on a network is represented as a finite string of alternating nodes and links $v_1 e_1 v_2 e_2 v_3 \dots e_r v_{r+1}$, with the property that v_i and v_{i+1} are linked by e_i for $i = 1, 2, \dots, r$. We require that $r \geq 1$, so that a walk will to traverse at least one link in the network.

A family of walks Γ is identified by the associated usage function that measures the usage of edges by members in the family, i.e., the usage function for e in $\gamma \in \Gamma$ is a number $\mathcal{N}(\gamma, e)$ that determines the number of times γ traverses e. For simple walks we have $\mathcal{N}(\gamma, e) \in \{0, 1\}$.

Therefore, by stacking the usage of each γ into a matrix, we obtain the usage matrix $\mathcal{N}_{|\Gamma|\times m}$ associated to Γ.

2.2 Admissible densities and p-energy of a density

Let $\rho : E \to [0, \infty)$ be a density where we interpret $\rho(e)$ as a penalization or cost that the walk γ must pay for traversing link e once. We define the ρ-length of a walk γ as

$$\ell_\rho(\gamma) \triangleq \sum_{e\in E} \mathcal{N}(\gamma, e)\rho(e) \tag{2.1}$$

and the ρ-length of a family of walks Γ as

$$\ell_\rho(\Gamma) \triangleq \inf_{\gamma\in\Gamma} \ell_\rho(\gamma)\,. \tag{2.2}$$

A density ρ is admissible for Γ if

$$\ell_\rho(\Gamma) \geq 1$$

in other words $\forall\gamma \in \Gamma$, $\ell_\rho(\gamma) \geq 1$. The admissibility condition can be written in matrix notation as:

$$\mathcal{N}\rho \geq \mathbf{1}$$

We denote the set of admissible densities by $\mathrm{Adm}(\Gamma)$.

Given $\rho \geq 0$, we define the p-energy of density ρ as

$$\mathcal{E}_{p,w}(\rho) = \sum_{e\in E} w(e)\,\rho(e)^p\,. \tag{2.3}$$

2.3 p-Modulus

For $1 < p < \infty$, $\mathrm{Mod}_{p,w}(\Gamma)$ is defined as

$$\mathrm{Mod}_{p,w}(\Gamma) = \min_{\{\rho | \ell_\rho(\Gamma) > 0\}} \frac{\mathcal{E}_{p,w}}{\ell_\rho(\Gamma)^p} \tag{2.4}$$

In this dissertation, we work with an equivalent form of (2.4)[50]:

$$\mathrm{Mod}_{p,w}(\Gamma) = \min_{\{\rho | \mathcal{N}\rho \geq 1\}} \mathcal{E}_{p,w}(\rho) = \mathcal{E}_{p,w}(\rho^*), \tag{2.5}$$

Correspondingly, the modulus problem in equation (2.5) can be recast into a convex optimization formulation:

$$\begin{aligned} \underset{\rho}{\text{minimize}} \quad & \sum_{e \in E} w(e)\, \rho(e)^p \\ \text{subject to} \quad & \mathcal{N}\rho \geq 1 \end{aligned} \tag{2.6}$$

It was shown in[52] that the set of admissible densities $\mathrm{Adm}(\Gamma) = \{\rho \in \mathbb{R}^E_{\geq 0} | \mathcal{N}\rho \geq 1\}\}$, is a receding polyhedran since any family of walks Γ can be replaced with a finite subfamily of walks Γ' called *essential subfimily*, with $\mathrm{Adm}(\Gamma) = \mathrm{Adm}(\Gamma')$. In particular, (4.2) is a convex optimization problem and for $1 < p < \infty$, the energy is strictly convex and thus (4.2) has a unique solution. The existence of an extremal density ρ^* for $p \geq 1$ is proven in[51] Lemma 2.1. To simplify notation, the subscript w will be omitted unless needed.

2.3.1 Properties of p-modulus

Proposition 2.3.1. *For any finite network $\mathcal{G}$, the following properties hold:*

(a) **p-Monotonicity:** *The extremal densities satisfy $0 \leq \rho^*(e) \leq 1$ for all $e \in E$. Thus, for $1 \leq p \leq q$, we have $\mathrm{Mod}_q(\Gamma) \leq \mathrm{Mod}_p(\Gamma)$.*

(b) **Γ-Monotonicity:** *If $\Gamma' \subset \Gamma$, then $\mathrm{Mod}_p(\Gamma') \leq \mathrm{Mod}_p(\Gamma)$.*

(c) **w-Monotonicity:** *If w and w' are positive link weights with $w \leq w'$ then* $\mathrm{Mod}_{p,w}(\Gamma) \leq \mathrm{Mod}_{p,w'}(\Gamma)$.

(d) **Empty Family:** *If $\Gamma = \emptyset$, then* $\mathrm{Mod}_p(\Gamma) = 0$.

(e) **Countable Subadditivity:** *For any sequence $\{\Gamma_i\}_{i=1}^{\infty}$ of families of walks,*

$$\mathrm{Mod}_p\left(\cup_{i=1}^{\infty}\Gamma_i\right) \leq \sum_{i=1}^{\infty} \mathrm{Mod}_p\left(\Gamma_i\right).$$

(f) **Extension Rule:** *Given two families of walks, Γ and Γ', if for all $\gamma \in \Gamma$, there exists $\gamma' \in \Gamma'$ such that γ' is subwalk (subordinate) of γ, i.e., $\mathcal{N}(\gamma', e) \leq \mathcal{N}(\gamma, e)$ for every $e \in E$. Then* $\mathrm{Mod}_p(\Gamma') \geq \mathrm{Mod}_p(\Gamma)$.

(g) **Parallel Rule:** *Given two families Γ_1 and Γ_2, such that $\mathcal{N}(\gamma_1, e)\mathcal{N}(\gamma_2, e) = 0$ for every $e \in E$, $\gamma_1 \in \Gamma_1$ and $\gamma_2 \in \Gamma_2$. Then* $\mathrm{Mod}_p(\Gamma_1 \cup \Gamma_2) = \mathrm{Mod}_p(\Gamma_1) + \mathrm{Mod}_p(\Gamma_2)$.

Proof. For (a), see[51] Lemma 2.2 and Theorem 5.5. For (b), (d)–(f), see[50] Proposition 3.4 and Section 5.5.

To prove (c), note that w does not affect the admissible set $A(\Gamma)$. Moreover, for any $\rho \in A(\Gamma)$, $\mathcal{E}_{p,w}(\rho) \leq \mathcal{E}_{p,w'}(\rho)$.

For (g), since the statement is slightly different than in[50], we provide a proof. By (e), we know $\mathrm{Mod}_p(\Gamma) \leq \mathrm{Mod}_p(\Gamma_1) + \mathrm{Mod}_p(\Gamma_2)$.

Let E_i be the set of links in E such that $\mathcal{N}(\gamma, e) \neq 0$ for some $\gamma \in \Gamma_i$. Note that $E_1 \cap E_2 = \emptyset$ by hypothesis. Given $\rho \in A(\Gamma)$, define $\rho_i = \rho \cdot \mathbf{1}_{E_i}$ for $i = 1, 2$, where $\mathbf{1}_{E_i}$ is the indicator function for E_i. Then $\rho_i \in A(\Gamma_i)$ for $i = 1, 2$, and $\mathcal{E}_p(\rho) \geq \mathcal{E}_p(\rho_1) + \mathcal{E}_p(\rho_2)$. Taking the infimum of both sides results in $\inf_{\rho \in A(\Gamma)} \mathcal{E}_p(\rho) \geq \inf_{\rho_1 \in A(\Gamma_1)} \mathcal{E}_p(\rho_1) + \inf_{\rho_2 \in A(\Gamma_2)} \mathcal{E}_p(\rho_2)$. Therefore, $\mathrm{Mod}_p(\Gamma) \geq \mathrm{Mod}_p(\Gamma_1) + \mathrm{Mod}_p(\Gamma_2)$. □

2.4 Basic families of walks; connecting, via and loop

In this section, we introduce three basic families of walks that will be fundamental later.

2.4.1 Connecting families

The family of connecting walks $\Gamma(A, B)$ is comprised of all walks that start on $A \subset V$ and end on $B \subset V \setminus A$ in the network G. We will often abbreviate $\text{Mod}_2(\Gamma(A, B))$ simply by writing $\text{Mod}_2(A, B)$ or $\text{Mod}_2(s, t)$ if $A = \{s\}$ and $B = \{t\}$.

On undirected networks, 2-Modulus of connecting families is the same as effective conductance, as described in[55] and in the following we show that modulus can be calculated analytically in undirected networks.

Formula for $\text{Mod}_2(a, b)$ in undirected networks

Let $\mathbb{F}$ be the set of all unit flows $\mathbb{f} : E \to \mathbb{R}$ that satisfy Kirchoffs node law and pass through a network G from a to b'. Namely for $v \in V$

$$(\nabla . \mathbb{f})(v) = \begin{cases} 1 & v = a \\ -1 & v = b \\ 0 & \text{o/w} \end{cases}$$

corresponds to the injected currents at each node. The energy of $\mathbb{f}$ is

$$\text{Energy}(\mathbb{f}) \triangleq \sum_{e \in E} \mathcal{R}(e)\mathbb{f}(e)^2$$

where $\mathcal{R}(e) = \frac{1}{w(e)}$ is the resistance of edge e. A unit current flow $\mathbb{i} \in \mathbb{F}$ is a unit flow that also satisfies Ohm's law, i.e., there is a function $\mathbb{V} : V \to \mathbb{R}$ (called a potential) such that for every edge (a, b):

$$\mathcal{R}(a, b)\mathbb{i}(a, b) = \mathbb{V}(b) - \mathbb{V}(a).$$

Let $\mathbb{U} : V \to \mathbb{R}$ be a vertex potential function. We can redefine the densities as the gradient of $\mathbb{U}$, i.e., for the edge $e = \{v, w\}$

$$\rho_{\mathbb{U}}(e) = |\mathbb{U}_u - \mathbb{U}_w|$$

Thus the admissibility condition for walks from a to b converts to $\mathbb{U}(a) = 0,\ \mathbb{U}(b) = 1$, and the 2-energy defined in (4.3) with $\rho_{\mathbb{U}}(e)$ is

$$\text{Energy}(\rho_{\mathbb{U}}) = \sum_{e \in E} \rho_{\mathbb{U}}(e)^2$$

assuming each edge has a unit resistance and substituting $\mathbb{U}$ by $\frac{\mathbb{V}}{\mathcal{R}_{\text{eff}}(a,b)} + C$, where $\mathbb{V}$ is the electric potential when a unit current flow $\mathbb{i} \in \mathbb{F}$ is passing through the network with source a and sink b and the effective resistance between a and b is $\mathcal{R}_{\text{eff}}$. By Thompson's principle, $\mathbb{i} \in \mathbb{F}$ is the minimizer of the energy function of all unit flows, i.e.,

$$\sum_{e \in E} \mathbb{i}(e)^2 = \min_{\mathbb{f} \in \mathbb{F}} \sum_{e \in E} \mathbb{f}(e)^2 = \mathcal{R}_{\text{eff}}(a, b)$$

Therefore,

$$\text{Mod}_2(a, b) = \min_{\substack{\mathbb{U}_a = 0 \\ \mathbb{U}_b = 1}} \rho_{\mathbb{U}}^T \rho_{\mathbb{U}} = \frac{1}{\mathcal{R}_{\text{eff}}(a, b)}. \tag{2.7}$$

By Kirchhoff's law of current conservation:

$$\sum_j A_{i,j}(\mathbb{V}_i - \mathbb{V}_j) = (\nabla . \mathbb{i})(i)$$

where $A = [a_{ij}] \in \mathbb{R}^{N \times N}$ is the adjacency matrix of G, with $a_{ij} = 1$ if and only if $i, j \in E$. In matrix form:

$$L\mathbb{V} = \mathbb{I} \tag{2.8}$$

where L is the Laplacian matrix of G and $\mathbb{I} = \nabla . \mathbb{i}$. Because $\mathbb{V}$ is defined up to an additive and the nullspace of L is along the constant vector, we ground an arbitrary node k and thus reduce L by removing kth row and column denoted by ${}^kL$ (see Figure 2.1). Now we can find solve (2.8):

$${}^k\mathbb{V} = ({}^kL)^{-1} \ {}^k\mathbb{I}.$$

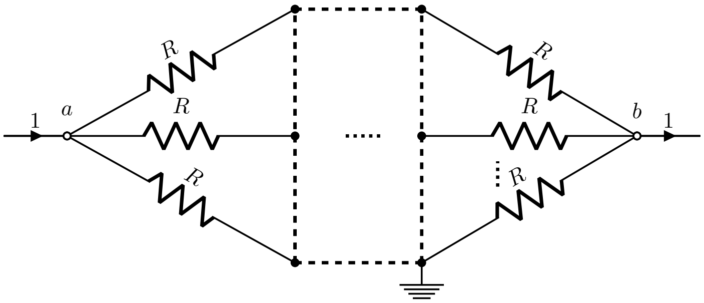


Figure 2.1: $\text{Mod}_2(a, b)$ represents the effective conductance between nodes a and b.

we denote $({}^kL)^{-1}$ by $\mathcal{G}$ (reduced conductance matrix) and obtain effective resistance between nodes a and b is

$$\begin{aligned}\mathcal{R}_{\text{eff}}(a, b) &= {}^k\mathbb{V}_a - {}^k\mathbb{V}_b \\ &= \mathcal{G}_{a,a} + \mathcal{G}_{b,b} - 2\ \mathcal{G}_{a,b}\end{aligned} \tag{2.9}$$

and from (2.7):

$$\text{Mod}_2(a, b) = (\mathcal{G}_{a,a} + \mathcal{G}_{b,b} - 2\mathcal{G}_{a,b})^{-1} \tag{2.10}$$

2.4.2 Family of walks visiting a set of intermediate nodes, "via" family

Another interesting family of walks is the *via family* $\Gamma_{via}\,(A, B; C)$[50], which represents the family of all walks that start from a set of nodes A, visit another set $C \subset V \setminus A$, and end on nodes $B \subset V \setminus (A \cup C)$.

By the extension property of modulus in Proposition 2.3.1, since $\Gamma(A, B; C)$ is a subordinate of $\Gamma(A, B)$, we have:

$$\text{Mod}_2(A, B) \geq \text{Mod}_2(A, B; C)$$

2.4.3 Family of loops

A walk $\gamma = v_1 v_2 v_3 \ldots v_r$, is a *simple loop* if the nodes v_i are all distinct, except that $v_r = v_1$. We call $\mathcal{L}$ the family of all loops in G. Other possible loop families are loop families rooted at a given node v or link e; we write $\mathcal{L}^v$ or $\mathcal{L}^e$ in that case.

For example, if G is a tree, $\mathrm{Mod}_p(\mathcal{L}) = 0$ by Property (d) above; if G is an unweighted complete graph, then $\mathrm{Mod}_p(\mathcal{L}) = \frac{1}{3^p}\binom{n}{2}$.

2.5 Interpreting modulus as a measure of the richness of a family of walks

The properties of modulus allow quantification of the richness of various family of walks, i.e., a family with many short walks has a larger modulus than a family with fewer and longer walks. In particular, Γ-monotonicity and subadditivity define a notion of *capacity* on the set of walks in a network.

In order to measure the richness of a family of walks, we want to balance the number of different walks with relatively little overlap and how short their lengths are. For example, in a connecting family of walks that connects two sets of nodes, we want to value many short walks. $\mathrm{Mod}_p(\Gamma(a,b))$ provides this measure, and by varying the values of p more emphasis can be placed on properties such as the number of walks or their length and bottlenecks, see[51]. On undirected networks, when $p = 2$ and the family Γ is the connecting family between two nodes, then $\mathrm{Mod}_2(\Gamma)$ recovers effective conductance. Therefore, we primarily restrict ourselves to $p = 2$ due to its physical interpretations and computational advantages.

Moreover, we will include families Γ that are not connecting and we can address networks that are directed.

For example, in Figure 2.2, Γ_0 is the connecting family between blue node s and orange node t. Here the networks are directed. Comparing Figure 2.2(a) to Figure 2.2(b), we see that every walk from s to t in the former contains a subwalk in the latter, thus the modulus increases by the extension rule (Proposition 2.3.1 (f)). In Figure 2.2(c), the weight of a link

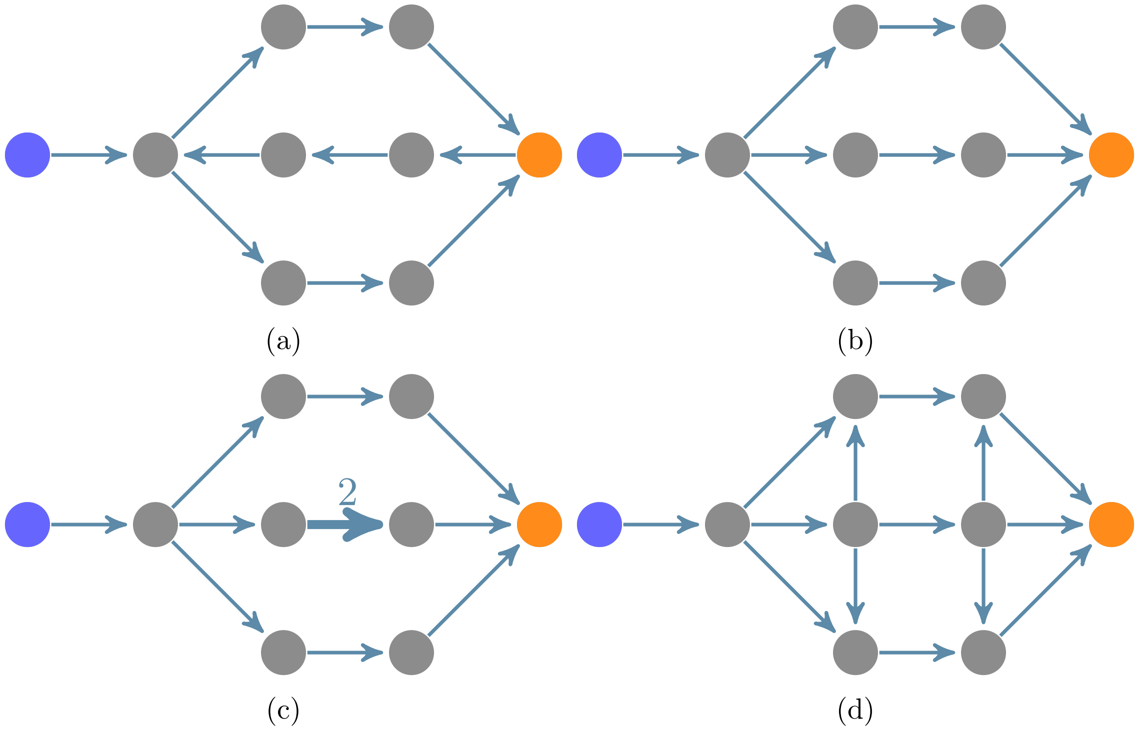


Figure 2.2: 2-Modulus of the family of connecting walks from a source node (blue node) to the target node (orange node); all links have weights 1, except one link in the graph (c). When the family is enriched, modulus increases, i.e., in (a) $\mathrm{Mod}_2(s,t) = 0.4$, (b) $\mathrm{Mod}_2(s,t) = 0.5$, (c) $\mathrm{Mod}_2(s,t) = 0.516$, (d)$\mathrm{Mod}_2(s,t) = 0.517$

s doubled to 2 and modulus increases as it must by w-monotonicity (Proposition 2.3.1 (c)). Figure 2.2(d) differs from Figure 2.2(b) in that the number of walks is higher than before, and modulus increases, demonstrating Γ-monotonicity (Proposition 2.3.1 (b)). The comparison between Figures 2.2(c) and 2.2(d) is more subtle; the relationship between the moduli is nontrivial since none of the monotonicity properties apply.

In another example, we want to measure the richness of a family of loops by balancing the number of different loops with relatively little overlap vs. how many short loops there are in the family.

We demonstrate this in Figure 2.3. For the square in Figure 2.3(a), the family $\mathcal{L}$ consists of a single loop, hence $\text{Mod}_2(\mathcal{L}) = 0.25$. In Figure 2.3(b), the weight of one link is doubled and modulus increases to $\text{Mod}_2(\mathcal{L}) = 0.285$, as it must, by w-monotonicity (Property (c)). The network in Figure 2.3(c) has more loops than the one in Figure 2.3(a) and modulus increases to $\text{Mod}_2(\mathcal{L}) = 0.5$, demonstrating $\mathcal{L}$-monotonicity (Property (b)). Comparing Figure 2.3(c) to Figure 2.3(d), we see that they have the same number of loops, but in (d) they are longer and thus the modulus decreases to $\text{Mod}_2(\mathcal{L}) = 0.455$.

2.6 Dual formulation for 2-modulus

For $p = 2$ the modulus problem in (4.2) is (for simplicity of algebra, we assume $w \equiv 1$)

$$\begin{aligned} \underset{\rho}{\text{minimize}} \quad & \sum_{e\in E} \rho^T \rho \\ \text{subject to} \quad & \mathcal{N}\rho \geq 1 \end{aligned} \tag{2.11}$$

We consider the Lagrangian for (2.11):

$$L(\rho, \lambda) = \rho^T \rho - \lambda^T \left(\mathcal{N}^T \rho - \mathbf{1} \right), \tag{2.12}$$

where $\lambda \in \mathbb{R}^{\Gamma}_{\geq 0}$ is the Lagrange multipliers. It is easy to show that $\rho = \mathbf{1}$ is an interior point for the feasible region of (2.11), thus by Slater's condition strong duality holds[62]. Minimizing

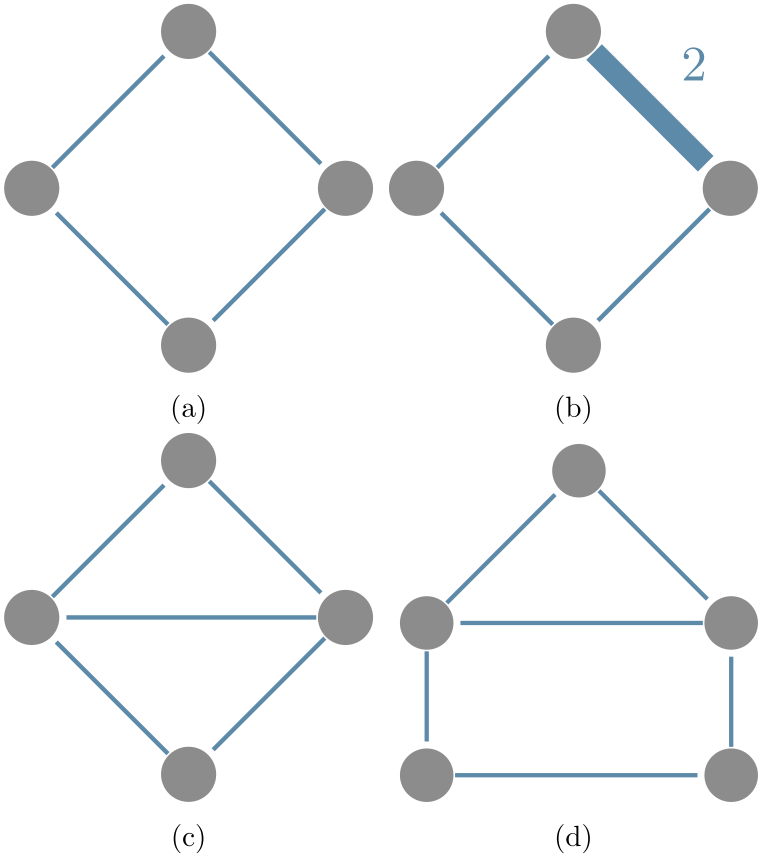


Figure 2.3: Loop Modulus for some networks demonstrating how modulus can quantify the richness of loops, a) $\text{Mod}_2(\mathcal{L}) = 0.25$ b) Weight of a link is doubled, modulus increase by w-monotonicity: $\text{Mod}_2(\mathcal{L}) = 0.285$ c) Increasing number of short loops the modulus increases by $\mathcal{L}$-monotonicity: $\text{Mod}_2(\mathcal{L}) = 0.5$. d) Loops are longer than (c) and modulus decreases: $\text{Mod}_2(\mathcal{L}) = 0.455$.

L in ρ gives

$$\rho^*(e) = \frac{1}{2} \sum_{\gamma \in \Gamma} \lambda^*(\gamma) \mathbb{1}_{e \in \gamma}, \tag{2.13}$$

and the dual problem:

$$\max_{\lambda \geq 0} \left(\lambda^T \mathbf{1} - \frac{1}{4} \lambda^T C \lambda \right). \tag{2.14}$$

where C is the *overlap matrix.* Namely for simple walks,

$$C(\gamma_i, \gamma_j) = \sum_{e \in E} \mathcal{N}(\gamma_i, e) \mathcal{N}(\gamma_j, e) = |\gamma_i \cap \gamma_j|$$

measures the overlap of two walks, i.e., number of edges in common between them.

From the KKT conditions, a pair $(\rho^*, \lambda^*) \in \mathbb{R}^E \times \mathbb{R}^\Gamma$ is optimal for the primal (2.11) and the dual problem (2.14), if and only if

- Primal-dual feasibility: $\rho^* \in \mathrm{Adm}(\Gamma)$ and $\lambda^* \geq \mathbf{0}$

- Complementary slackness: $\forall \gamma \in \Gamma$

$$\lambda^*(\gamma) \left(1 - \sum_{e \in E} \mathcal{N}(\gamma, e) \rho^*(e) \right)$$

- Stationarity:

$$(\nabla_\rho \mathcal{L})(\rho^*, \lambda^*) = 0$$

If we know the minimal subfamily of Γ denoted by Γ' then C is invertible and $\lambda(\gamma'(e)) > 0$, we can analytically solve the dual problem by

$$\nabla_\lambda \left(\lambda^T \mathbf{1} - \frac{1}{4} \lambda^T C \lambda \right) = 0$$

obtaining $\lambda^* = 2C^{-1}\mathbf{1}$ and from strong duality $\mathrm{Mod}_2(\Gamma) = \mathbf{1}^T C^{-1} \mathbf{1}$.

Algorithm 1 Approximating densities for $\mathrm{Mod}_2(\Gamma)$ with tolerance $0 < \epsilon_{\mathrm{tol}} < 1$

```
$\rho \leftarrow 0$; $\rho_0 \leftarrow \mathbf{1}$
$\Gamma' \leftarrow \emptyset$
$\gamma \leftarrow \mathit{Shortest}(\rho_0)$
while $\ell_\rho(\gamma) \leq 1 - \epsilon_{\mathrm{tol}}$ do
    $\Gamma' \leftarrow \Gamma' \cup \{\gamma\}$
    $\rho \leftarrow \mathrm{argmin}\{\mathcal{E}_2(\rho) : \mathcal{N}\rho \geq \mathbf{1}\}$
    $\gamma \leftarrow \mathit{Shortest}(\rho)$
end while
```

2.7 Approximating the modulus

The numerical results in this dissertations are produced by a Python implementation of the simple algorithm described in[50]. This algorithm exploits the Γ-monotonicity (Property (b)) of the modulus by building a subset $\Gamma' \subseteq \Gamma$ so that $\mathrm{Mod}_2(\Gamma') \approx \mathrm{Mod}_2(\Gamma)$ to a desired accuracy[50] Theorem 9.1. In short, the algorithm begins with $\Gamma' = \emptyset$, for which the choice $\rho \equiv 0$ is optimal, and repeatedly adds violated constraints to Γ', recomputing the optimal ρ each time. The algorithm terminates when all constraints are satisfied to a given tolerance (Algorithm 1).

The two key ingredients for implementing this algorithm are a solver for the convex optimization problem (4.2) and a method for finding violated constraints, i.e., finding shortest walks with ρ-length less than one. In our implementation, the optimization problem is solved using an active set quadratic programming solver[63] and the violated constraint search algorithm varies based on the walk family. For example, shortest walk between a pair of nodes can be found using Dijkstra's algorithm.

Although simple, this algorithm is adequate for computing the modulus in the most of the examples presented here, on a Linux operating computer with Intel core i7 (and 2.80 GHz base frequency) processor, for example. More advanced parallel primal-dual algorithms can be developed to treat modulus computations on larger networks.

Chapter 3

Network measures based on shell modulus

The modulus of a family of walks quantifies the richness of the family by favoring many short walks over fewer longer ones. In this chapter, we investigate various families of walks in order to introduce new measures for quantifying network properties using modulus. The proposed new measures are compared to other known quantities such as current-flow closeness centrality, out-degree centrality, and current-flow betweenness centrality. Our proposed method is based on walks on a network, and therefore will work in great generality. For instance, the networks we consider can be directed, multi-edged, weighted, and even contain disconnected parts. Examples are provided to show the effectiveness of our measures.

In this chapter, we explore the versatility of modulus of families of walks, demonstrating that it provides a powerful approach to the study of networks. We describe different problems that can be handled by various classes of families of walks. Furthermore, we propose measures based on these families that can be applied in a general framework, handling directed or undirected, weighted or unweighted, and disconnected networks, while the amount of information extracted from a network can be adjusted with high accuracy. We apply the proposed measures to problems such as detecting influential parts of networks, ranking most important nodes in networks, and mitigating the spread of infection in networks. These measures capture the importance of a node on a network by considering only parts of the network most strongly influenced by the node. Thus, the computation of these proposed measures can be done in a decentralized manner, allowing an efficient parallel implementation for large networks.

The chapter is organized as follows. We define our proposed measures in detail and compare them to different conventional measures, and apply these new measures in various examples and applications.

3.1 Closeness Centrality

There are many centrality measures for evaluating the importance of nodes in networks. The simplest measure is degree, which considers immediate neighbors. Degree ignores the rest of the network and hence cannot be a proper measure for evaluating the importance of a node in the whole network. Another centrality measure is closeness centrality, which measures the closeness of a node in a network by computing the reciprocal of the sum of shortest-path distances from the given node to all other nodes[64]:

$$C_c(v) = \frac{1}{\sum\limits_{u \in V} d(v,u)} \tag{3.1}$$

where $d(v,u)$ is the distance between node v and u.

As described in[65], any reasonable centrality measure should increase when more links are added to the network. For connecting families of walks, modulus does have this behavior because of Proposition 2.3.1 (b). The main purpose of this chapter is to introduce new centrality measures based on modulus of families of walks. For instance, a simple first attempt is to define

$$C\left(v\right) := \sum_{i \in V} \mathrm{Mod}_2\left(v,i\right) \tag{3.2}$$

where $\mathrm{Mod}_2\left(v,i\right)$ is the 2-Modulus of the family of all walks from node v to node i.

Note that modulus of connecting families is a measure of proximity rather than a distance, so this is roughly analogous to the classical measure (3.1).

However, in order to obtain this centrality measure for all nodes, we need to compute 2-Modulus, n^2 times. We propose a more efficient measure based on modulus below in Section 3.3.

Remark 3.1.1. Note that in a directed network we could also define $C_{in}(v) = \sum_{i\in V} \text{Mod}_2(i,v)$, which measures the richness of walks that are reaching v, and $C_{out}(v) = \sum_{i\in V} \text{Mod}_2(v,i)$, which computes the influence of node v over the network. In this chapter we will mostly be concerned with the latter measure.

3.2 Betweenness centrality

Betweenness centrality evaluates the prominence of v in the transmission of information, disease, signals, etc., between pairs of nodes.

A popular betweenness centrality for a node v is defined by the fraction of shortest paths between pairs of nodes that pass through the node v:

$$C_B(v) = \sum_{s\neq t\neq v\neq s} \frac{\sigma_{st}(v)}{\sigma_{st}} \tag{3.3}$$

where σ_{st} is the number of shortest paths between node s and node t and $\sigma_{st}(v)$ is the size of the subset of such shortest paths that visit v[66].

An analogous measure of betweenness centrality could be defined using *via* Mod, which computes the richness of walks between s and t that pass through node v (see Section 2.4.2):

$$BC(v) := \sum_{s\neq t\neq v\neq s} \frac{\textit{via}\,\text{Mod}_2(s,t;v)}{\text{Mod}_2(s,t)} \tag{3.4}$$

A naive implementation of this formula requires $|V|^3+|V|^2$ modulus computations. Although not much is currently known about the computational complexity of modulus of general families of walks, in our experience computing the modulus of *connecting* and *via* families is reasonably fast due to the applicability of Dijkstra's algorithm in these cases (see[50]) and the fact that these families contain very few important walks (see[61]). Nevertheless, for efficiency it is desirable to reduce the number of modulus computations and, therefore, we propose the following more efficient measures.

3.3 Efficient measures for centrality based on modulus

The centrality measures introduced above consider all pairs and triples of nodes, which can be infeasible for large networks. In applications the entire scope of network data cannot always be obtained, and even if it is acquired, such an extensive volume of data is sometimes unnecessary. Each node v will influence some nodes more than others. Therefore, we restrict our analysis of centrality for v to a portion of network aound v. The following section describes a technique used for this purpose.

3.3.1 Shell-Centrality

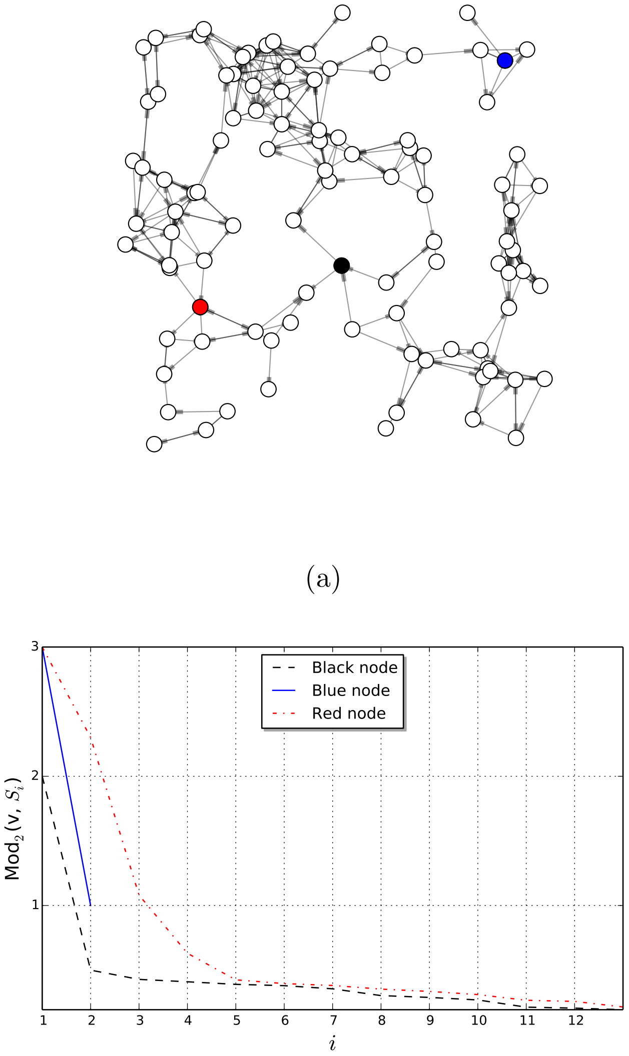
For a node $v \in V$, $S\left(v,k\right)$ is the set containing nodes y such that $\{y \in V : d\left(y,v\right) = k\}$), namely all the nodes with discovery time k. We call this the k-th *shell* around v. If the context is clear, we simply write S_k.

We are interested in the 2-Modulus of all walks from node v to the shell S_k, which according Section 2.4.1, $\text{Mod}_2\left(v,S_k\right)$ is a measure of conductance between node v and the set S_k. Note that $\text{Mod}_2\left(v,S_1\right)$ is equal to the out-degree of v, corresponding to the influence of v on its immediate neighbors. (The extremal density ρ^* in this case gives value 1 to every out-link from v, and value 0 to every other link.)

The modulus from v to the shell S_k is a measure of the importance of node v out to radius k in the network. When a node has a persisting effect on larger and larger radii, it will have an overall greater closeness centrality (Figure 3.1).

Our proposed measure for the centrality of a node v, which we call *shell centrality*, is

$$\mathcal{C}_{\text{shell}}\left(v\right) := \sum_{i=1}^{d(v)} \text{Mod}_2\left(v,S_i\right) \tag{3.5}$$



(a)

(b)

Figure 3.1: (a) A directed network; (b) Plot of $\text{Mod}_2\,(v, S_i)$ with various shells i, given three different nodes for the network in (A). The Red node maintains its influence over the network and reaches farther, while the Blue node can influence only its first two shells. Out-degree of Blue is more than Black and it is more influential on shells 1 and 2 but Black node has influence over a larger part of the network, therefore Black node has higher centrality than Blue and Red is the more central than both Blue and Black. Link directions are shown by thicker stubs.

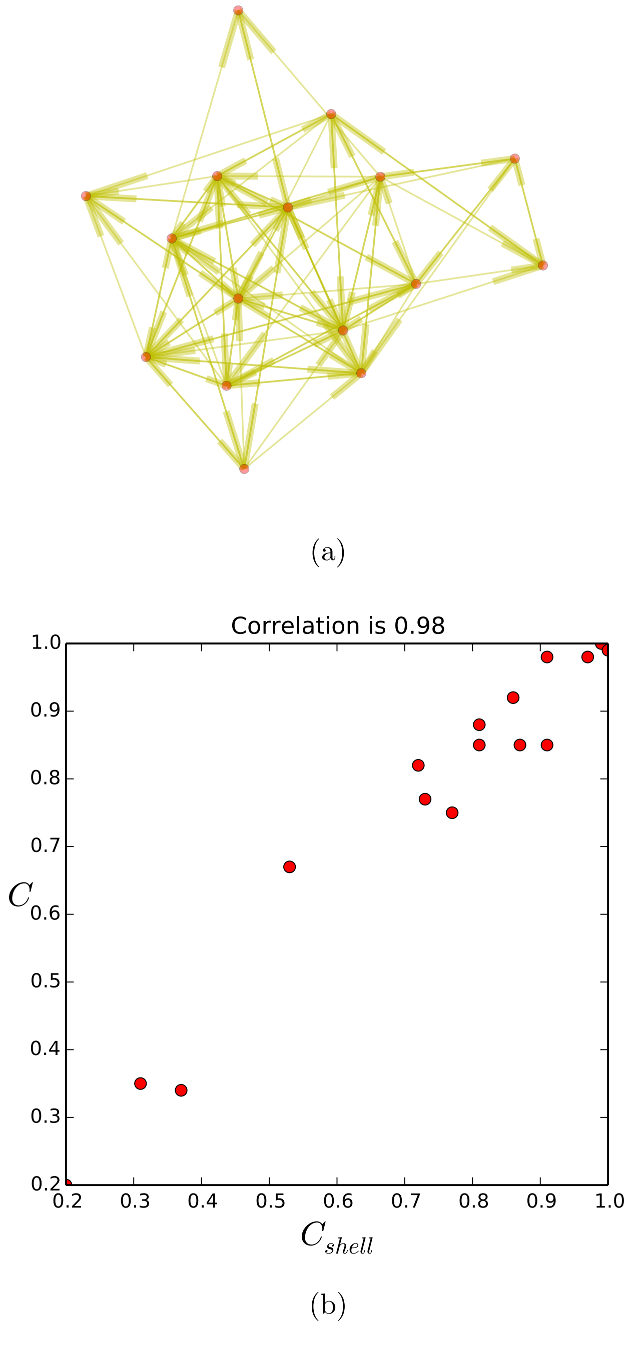
where $d(v) \in \{1, 2, \cdots, \epsilon(v)\}$ is a cutoff to be determined later, that can vary from node to node. The largest $d(v)$ is called the eccentricity of v, i.e., $\epsilon(v)$, is the maximum hop-length between v and any other node in $\mathcal{G}$.

Example 3.3.1. *In order to compare the proposed measure with the exact expression of C in (3.2), we compute C and $\mathcal{C}_{shell}$ for a weighted and directed network found in*[4] *(Figure 3.2). Each node in this network represents a rhesus monkey, and links between nodes represent observed grooming behavior. The direction of each link indicates the act of grooming. The correlation between C and $\mathcal{C}_{shell}$ is 98%, thereby demonstrating that the measure obtained almost the same results as (3.2) with less computation costs. In Figure 3.2, the centrality value of nodes is normalized by their maximum value, therefore for a node.*

Example 3.3.2. *In this example, we compute normalized $\mathcal{C}_{shell}$ with (3.5) for four weighted and directed networks, each differing from the previous one by one link. In Figure 3.3, the size of each node is scaled by its centrality, as computed by (3.5). The centrality value can be observed inside the nodes.*

Changes of nodes centralities provide interesting, and in some cases significant, assessments of a node's influence on the entire network. In Figure 3.3(a), the yellow node is the most central node, thereby influencing the entire network more than any other node. For the network in Figure 3.3(b), the direction of the link from the yellow node to the magenta node has been changed and the centrality updated, with the result that the centrality of the magenta node is now the highest. As shown in Figure 3.3(b), the white and magenta nodes have identical (weighted) out-degree centrality, but the white node cannot influence the network on its left side. If a link is added from white to magenta (Figure 3.3(c)), then the white node becomes the most central node. The addition of another link with a new node at the tail of the network, as shown in Figure 3.3(d), only changes centrality of the cyan node, while centrality of the other nodes stays almost constant. With out-degree centrality the cyan node would be the most central node.

Example 3.3.3. *As mentioned, for large networks acquisition or consideration of all data is not always possible. Therefore, there is a trade-off between the amount of information*



(a)

(b)

Figure 3.2: Correlation between normalized closeness centralities measured by (3.2) and (3.5) for the Rhesus Network in[4]. Link directions are shown with thick link heads.

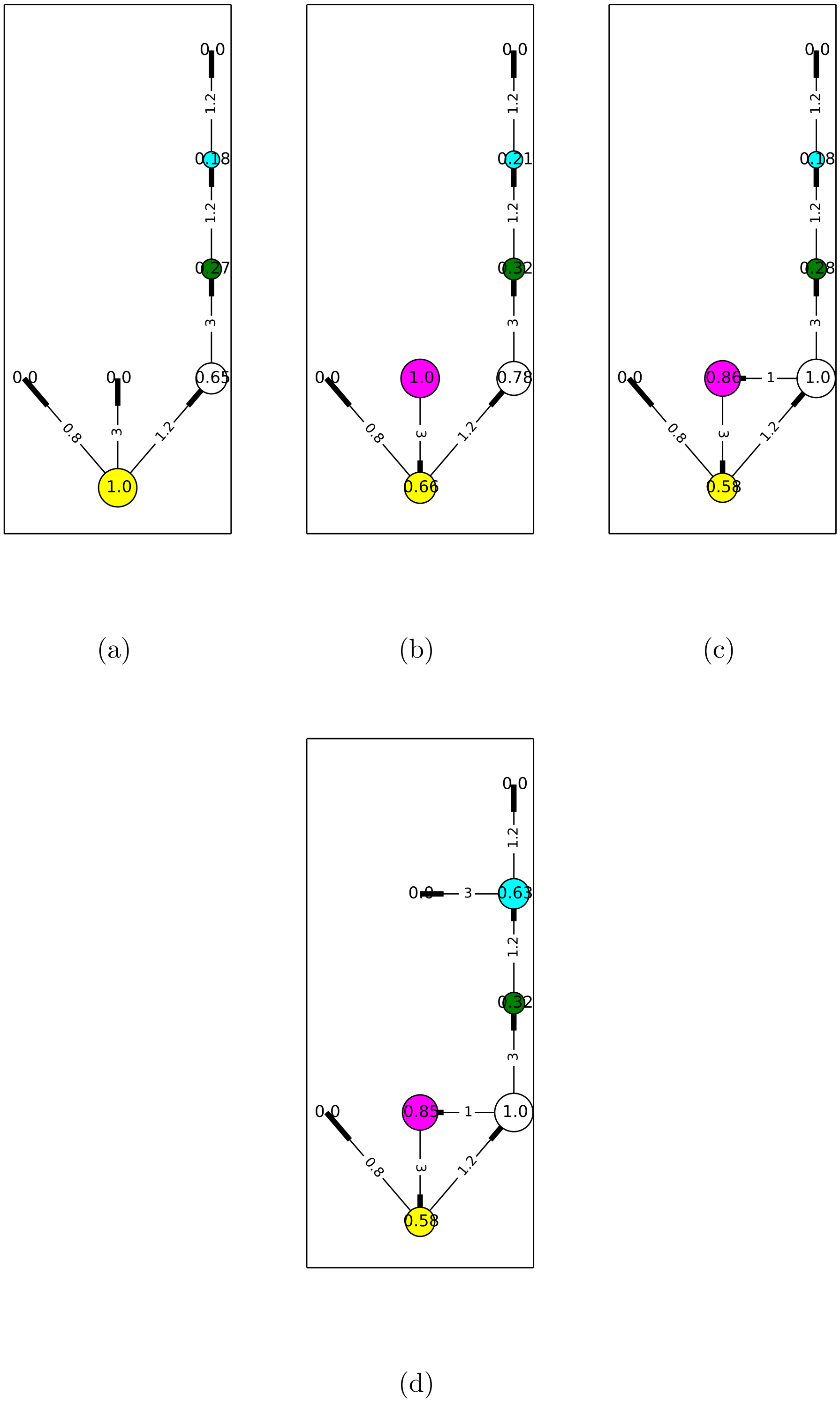
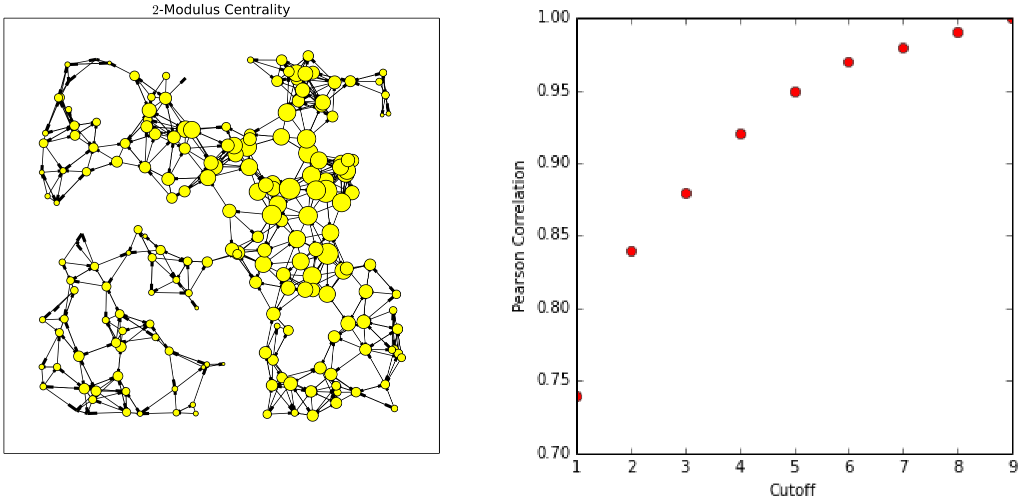


Figure 3.3: Closeness centrality measured for nodes in a directed and weighted network; weights are shown on links and directions are shown by thicker stubs. The size of each node is scaled by its centrality.



(a) (b)

Figure 3.4: (a) Random directed network; node size is enlarged with respect to its closeness centrality based on 2-Modulus centrality, (b) Correlation between exact 2-Modulus centrality and approximated one with varying cutoffs.

extracted and computational costs. For example, in Figure 3.4, we consider a random directed geometric network and plot the correlation between the centrality $\mathcal{C}_{shell}$ *computed with* $d(v) = \epsilon(v)$ *in (3.5), and the same centrality computed with different cutoffs* $d(v) = r$ *for various radii* r *of shells. By increasing these cutoff radii, we obtain increasingly better correlation, but after having reached* $r = 4$ *it seems that increasing the cutoff becomes unnecessary (Figure 3.4). This reflects the fact that, after* 6 *hops from each node in this network, the importance of the node starts to decay rapidly throughout the entire network and hence considering the first* $3 - 4$ *shells is enough for most applications.*

Remark 3.3.4. For a general network, it is difficult to predict the proper cutoff for a given node. In practice, we introduce a tolerance that is used to stop whenever $\text{Mod}_2(v, S_k)$ is less than a given value.

3.3.2 Betweenness centrality measure

Consider the visiting family of walks described in Section 2.4.2. Here, we introduce a modification of this family more well suited to the betweenness centrality measures defined in

Section 3.3.2. Let $A \subset V$ and let $c \in V \setminus A$. We define the *betweenness family* $\Gamma_b(A;c)$ as the family of walks originating at some node $a \in A$, visiting node c and then terminating at a node in $A \setminus \{a\}$. In other words,

$$\Gamma_b(A;c) = \bigcup_{a\in A} \Gamma_{via}(a, A \setminus \{a\}; c).$$

In the sequel, we will write $\text{Mod}_p\left(\Gamma_{via}\left(A, B; C\right)\right)$ as $via\,\text{Mod}_p\left(A, B; C\right)$ and $\text{Mod}_p\left(\Gamma_b\left(A; c\right)\right)$ as $\text{BMod}_p\left(A; c\right)$.

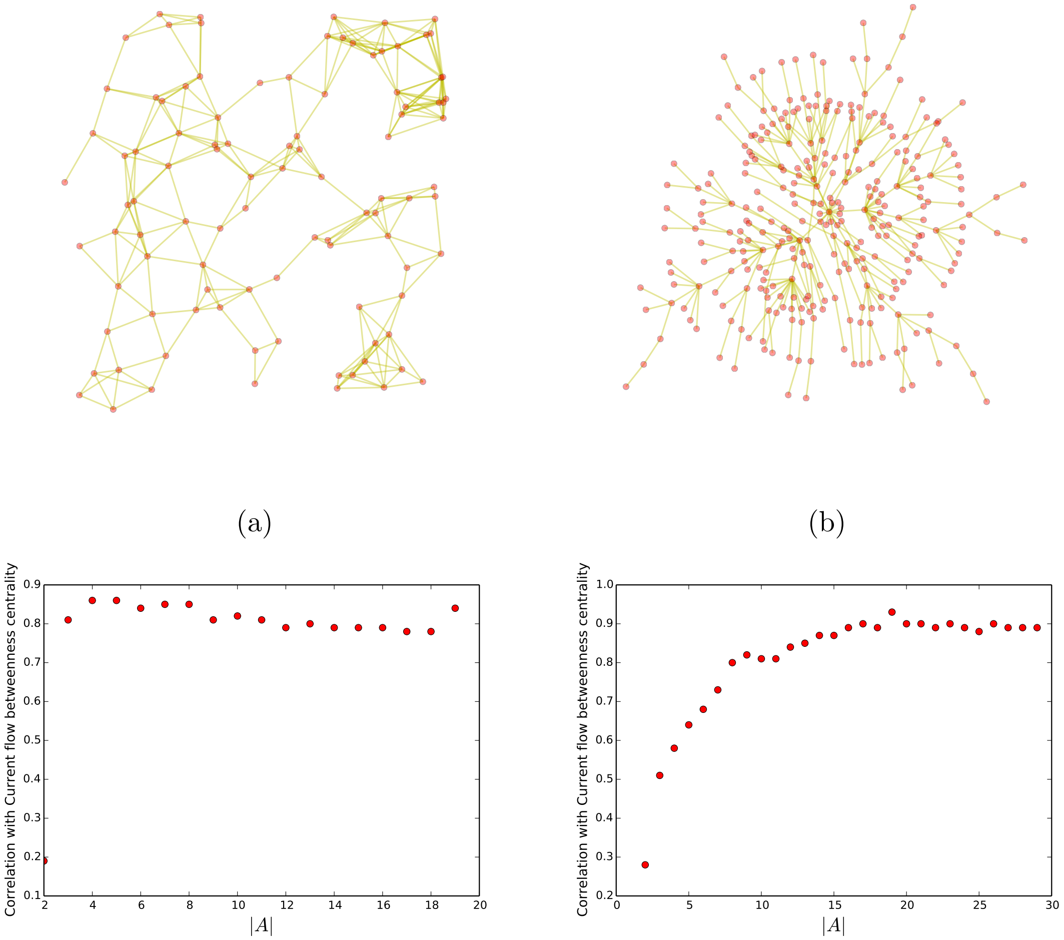
If walks begin from a node $a_i \in A$, visit node v and return to a different node $a_j \in A$ where $i \neq j$, we called this family $\Gamma_{via}\left(A, A, v\right)$. We choose A to be a proportion of the most central nodes using our centrality $\mathcal{C}_{\text{shell}}$ from (3.5). Then, we set

$$BC_{\text{shell}}\left(v\right) = via\,\text{Mod}_2\left(A, A, v\right) \tag{3.6}$$

Note that only one modulus computation is involved in (3.6) for each node. The number of nodes considered in A vary with the type of network, but BC_{shell} generally provides good results even when considering a handful of nodes. Also, if A happens to include all the neighbors of v, then BC_{shell} simply gives the degree of v.

Example 3.3.5. *We know 2-Modulus of a family of connecting walks between two nodes is equal to effective conductance between them. Thus, one might expect the betweenness measure (3.6) to be related to the well-known current-flow betweenness centrality (CFBC). However, the modulus-based centrality measure is more general in that it is not restricted to connected, undirected networks. In this example, we provide evidence that BC_{shell} and CFBC are linearly correlated by considering both a random geometric network and a random scale-free network.*

We calculate (3.6) for a geometric network[5] *and a scale-free network*[6] *as shown in Figure 3.5. We illustrate the correlation between well-known current-flow betweenness centrality*[7]*, and our measure of betweenness centrality BC_{shell} when the number of most central nodes chosen for the set A varies.*



(a)

(b)

(c)

(d)

Figure 3.5: (a) A geometric network[5], (b) A scale-free network[6], (c) Correlation between current flow betweenness centrality and proposed betweenness centrality computed with various number of servers for the geometric network in (a), (d) Correlation between Current flow betweenness centrality and proposed betweenness centrality computed with different number of nodes in the set A for the scale-free network in (b).

For the geometric network in Figure 3.5(a), there are only a few important nodes and they are scattered. So when the size of A increases past a certain level, A starts to include nodes on the periphery of the network and as shown in Figure 3.5(c) the correlation starts to decrease slightly.

In the scale-free network in Figure 3.5(b), central nodes are more accurately identifiable and they are more concentrated. So including more nodes in A leads to better and better correlation. However, a relatively small set A (here about 10% of the nodes in the network) can already give high correlation.

3.4 Further Results and Applications

In order to evaluate the effectiveness of our proposed measures (3.5) and (3.6), we compare them to various conventional network measures. First, we consider undirected, unweighted, and connected (simple) networks, and compare our measure $\mathcal{C}_{\text{shell}}$ to a well-known measure, current-flow centrality, demonstrating that they lead to similar results. Then, we consider general networks where current-flow centrality cannot be applied, and thus illustrate the advantages of using 2-Modulus centrality measures. Finally, we give two applications.

3.4.1 Undirected networks

For connected undirected networks there are numerous centrality methods[65]. In particular, the symmetry of the Laplacian matrix allows one to use measures such as effective conductance, also knows as current-flow closeness centrality (CFC)[7].

In order to evaluate the performance of $\mathcal{C}_{\text{shell}}$ in (3.5), we compare it to CFC in a simple geometric network with 60 nodes as shown in Figures 3.6(a) and 3.6(b).

These figures illustrates the measured proposed centrality and its correlation with current flow centrality. The correlation of these measures is 0.97, implying very similar rankings. This means that our measure is at least as good as CFC for simple networks.

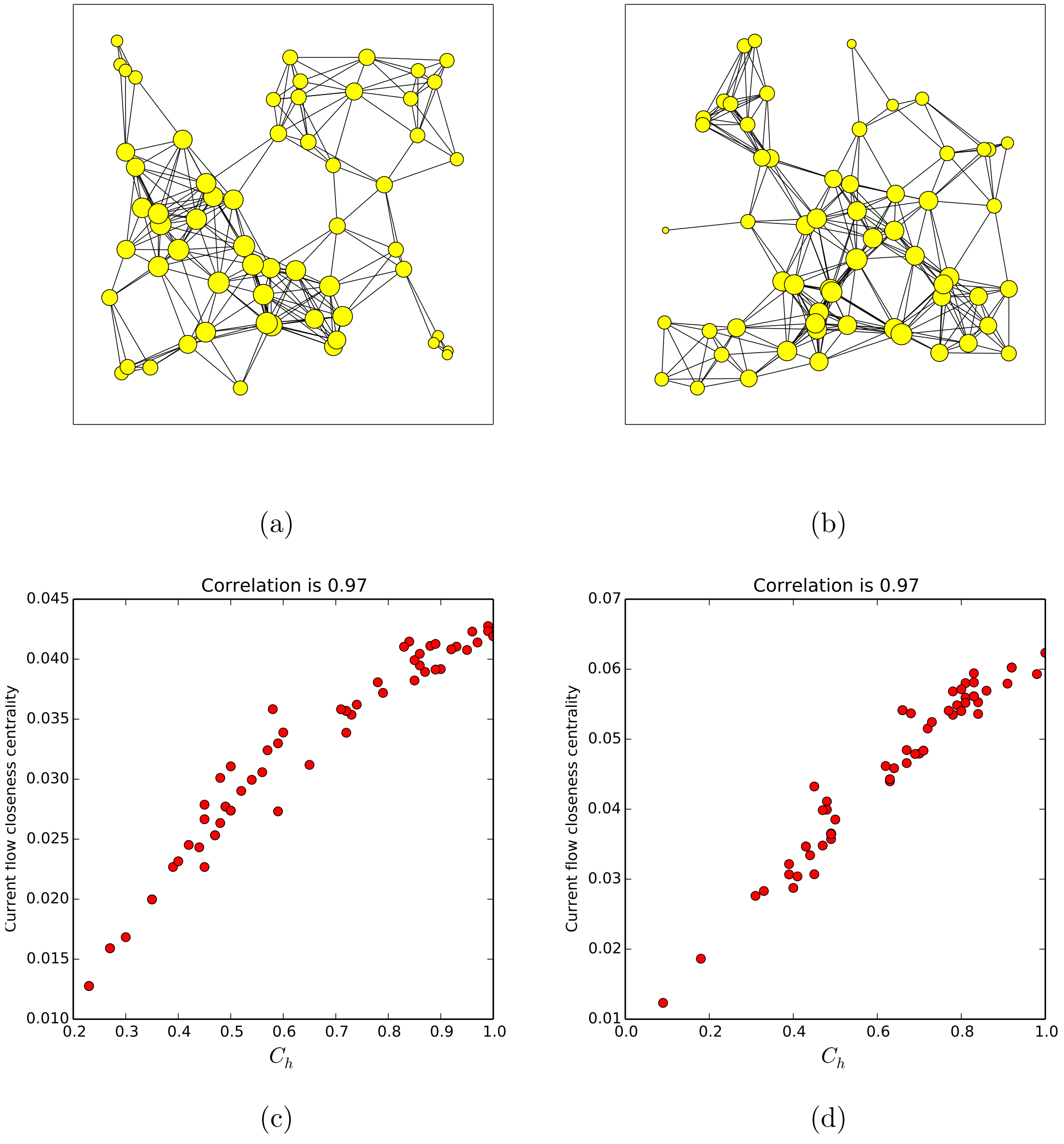


Figure 3.6: Closeness centrality measured with $\mathcal{C}_{\text{shell}}$ and correlation with current flow closeness centrality[7] for two networks (A) and (B) with node size. In both cases, (C) and (D), the correlation is 0.97.

3.4.2 Directed networks

There are fewer closeness centrality measures for directed networks compared to undirected networks, and most of the measures focus on local information of nodes. There are other measures that can be applied for directed networks, such as Pagerank and Katz centrality (for a good review, see[67]), but because they have some shortcomings in directed networks when there is a lack of mixing, we chose to compare our measure to out-degree centrality.

In Figure 3.7, we compare 2-Modulus centrality and out-degree centrality for two random directed networks, showing these centralities using the size of the nodes.

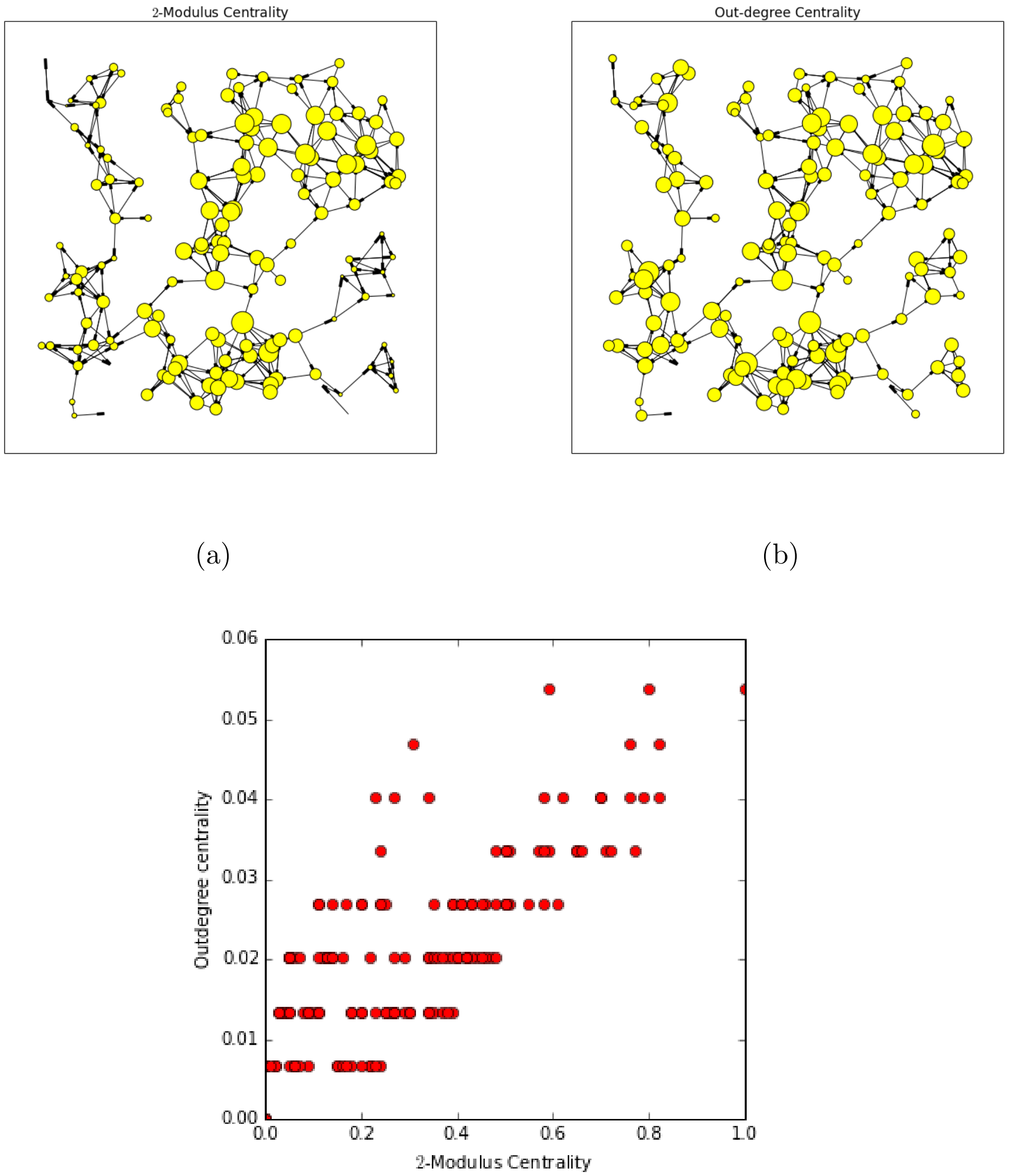
As depicted in these figures, out-degree centrality emphasizes the local importance of nodes, while 2-Modulus closeness centrality takes a broader perspective of the network. Consequently, nodes of the network that cannot reach most of the network have less importance in 2-Modulus centrality; however, in out-degree centrality nodes can have high centrality if they have high out-degree, as shown in Figure 3.7(c) in which nodes that have high out-degree centrality and small 2-Modulus centrality corresponded to nodes of the network that did not significantly influence the entire network.

3.4.3 Ranking of most influential nodes

Comparing nodes with a low number of heavily weighted links to nodes with a high number of more lightly weighted links, is usually a challenge for measures such as out-degree centrality[68]. However, 2-Modulus centrality does not have this problem, because it is not a local property. For instance, a heavily weighted link might lead to a smaller portion of the network.

Here we consider the network in[69] that consists of relationships between a group of 32 scientists. In this network, directed links are weighted by the number of sent messages between each pair of researchers.

Opsahl *et al.* ranked the nodes in this network with a centrality measure that upgraded out-degree centrality that can be tuned between the number of outward links and the sum of out-weights[68]. Since this centrality measure considers only links to the nearest neighbors,



(a)

(b)

(c)

Figure 3.7: 2-Modulus closeness centrality (A) and out-degree centrality (B) measured for a random directed networks. (C) 2-Modulus centrality and out-degree centrality. Each dot represents a node with x-axis as 2-Modulus centrality and y-axis as out-degree centrality.

it ignores most of the network structure. For example, ranking errors occur when a strong link is directed to a dead end in the network (or to an unimportant part of the network) but a weaker link is directed to important parts of the network.

In Table 3.1, we propose a new ranking performed based on 2-Modulus centrality with no concern for balancing between the number of paths and their weight strengths and with an eye on the node position in the network.

3.4.4 Suppressing epidemics

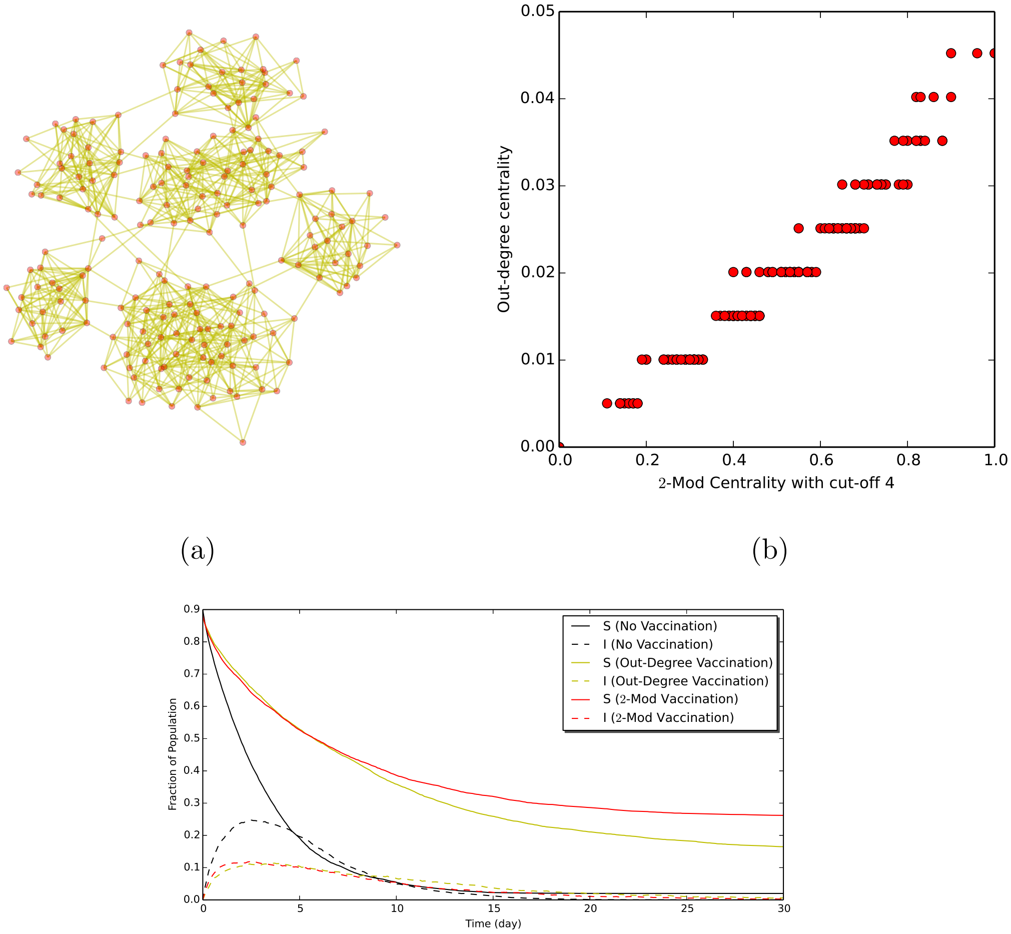
Detection of the most influential nodes is critical in some applications. Vaccination is commonly used to mitigate the spread of an infectious disease. However, it is not always possible to vaccinate the entire population. Therefore, determining the best sub-population to vaccinate is a challenge due to network complexity[2;70]. In this section, we show that the proposed measure can be efficiently used to vaccinate a fraction of highly central nodes, especially for directed networks with mesoscopic structure. A majority of real networks are formed by connecting clusters of sub-networks, such as communities. Each community contains its own structure and connects to others with a different structure. However, local measures, such as degree centrality, cannot capture these higher order structures.

In this study, we considered a random directed network consisting of clusters with internal Poisson degree distribution, which are connected to each other by another Poisson distribution[71]. In order to consider a directed version of these networks, we chose a direction for each link at random. Figure 3.8(a) shows a network generated in this way with 200 nodes and 8 modules. We compute 2-Modulus centrality with a cutoff of 4 and out-degree centrality. As presented in Figure 3.8(c)(b), nodes with identical out-degree centrality can have a different role in the network based on 2-Modulus centrality.

We consider an SIR (susceptible-infected-recovered) epidemic process on this network with infection rate $\beta = 0.5$ and recovery rate $\delta = .2$, starting with two initial infections. After vaccinating the first 50 nodes with the highest centrality in both measures, we ran several simulated epidemics (hundreds) with the parameters specified above and computed the

Table 3.1: Ranking of scientists in network EIES according to their 2-Modulus centrality and degree centrality scores.

Rank	2-Modulus Centrality (value)
1	LIN FREEMAN (1.0)
2	BARRY WELLMAN (0.78)
3	RUSS BERNARD (0.67)
4	LEE SAILER (0.55)
5	DOUG WHITE (0.51)
6	PAT DOREIAN (0.44)
7	SUE FREEMAN (0.33)
8	NICK MULLINS (0.21)
9	RON BURT (0.2)
10	RICHARD ALBA (0.19)
11	STEVE SEIDMAN (0.17)
11	AL WOLFE (0.17)
12	CAROL BARNER-BARRY (0.15)
13	JACK HUNTER (0.14)
13	MAUREEN HALLINAN (0.14)
14	PAUL HOLLAND (0.12)
15	DAVOR JEDLICKA (0.11)
15	JOHN BOYD (0.11)
16	PHIPPS ARABIE (0.08)
17	DON PLOCH (0.07)
18	MARK GRANOVETTER (0.05)
18	CLAUDE FISCHER (0.05)
19	JOEL LEVINE (0.04)
19	NAN LIN (0.04)
19	NICK POUSHINSKY (0.04)
19	CHARLES KADUSHIN (0.04)
20	JOHN SONQUIST (0.02)
21	BRIAN FOSTER (0.01)
21	EV ROGERS (0.01)
21	GARY COOMBS (0.01)
21	ED LAUMANN (0.01)
21	SAM LEINHARDT (0.01)



(a) (b)

(c)

Figure 3.8: (a) Random modular network, (b) Out-degree centrality and 2-Modulus centrality measured for each node in network (a), (c) Comparison of vaccination strategies based on 2-Modulus centrality and out-degree centrality for an SIR epidemic in network (A). The fraction of susceptible population (S) at the end of the outbreak for 2-Modulus centrality vaccinated people is larger than the fraction obtained by vaccinating using out-degree centrality.

average fraction of susceptible (not yet infected) individuals for each day of the outbreak[72]. As shown in Figure 3.8(c), 2-Modulus centrality pinpointed the most effective nodes better and allowed a more successful mitigation of the outbreak.

Chapter 4

Egocentric network centralities and general degree

Ego networks or also known as neighborhood networks are considered as samples of the complete network around egos that help to gain insights about the population and it is an exciting solution due to their flexible data collection and inexpensive computation costs. How to collect this samples in a way that significant conclusions can be made is a challenge due to the inherent structure of the network that can be lost in the ego centric data collection. In this chapter, we focus on the centrality measures in this kind of networks with considerations that justify the applications of them as scalable tools as a substitute for sociocentric methods.

The outline of the chapter is as following. First, we introduce our tools for analyzing families of connecting walks and redefine the popular effective conductance and its ego-centric version. Second, we discuss how to calculate this measures analytically and approximately. Third, we introduce general degree to incorporate higher order neighborhood in the simple degree measure. Proofs, examples and applications of these measures are postponed to supplemental materials.

4.1 From sociocentric measures to the egocentric counterparts

The concept of information centrality was first introduced in[73] and was later reinterpreted in terms of electrical conductance in[74]. Given a network $G = (V, E)$ and a node $a \in V$, the information centrality of a is defined as

$$\mathcal{C}_{\text{eff}}(a) := \sum_{b \neq a} \frac{1}{\mathcal{R}_{\text{eff}}(a, b)}. \tag{4.1}$$

where $\mathcal{R}_{\text{eff}}(a, b)$ is effective resistance distance between a and b in a resistance networks. Note that this measure considers every possible path that electrical current flow might take from a to an arbitrary sink b.

The situation can be clarified by introducing the notion of modulus of families of walks. This is a way of measuring the richness of certain families of walks on a network (and beyond, see[52;75]). Given two nodes a and b we may consider the connecting family $\Gamma(a, b)$ of all walks γ from a to b. Then, given edge density $\rho : E \to \mathbb{R}$ for $p \in [1, \infty]$, we define the p-modulus of Γ to be

$$\text{Mod}_p(\Gamma) \triangleq \min_{\ell_\rho(\Gamma) \geq 1} \text{Energy}_p(\rho) \tag{4.2}$$

Namely, we minimize the energy of candidate edge-densities ρ subject to the ρ-length of every walk in Γ being greater than or equal one, i.e., $\ell_\rho(\Gamma) \geq 1$. These densities can be interpreted as costs of using the given edge and then modulus is a constrained convex optimization problem that has a unique extremal density ρ^* when $1 < p < \infty$. The energy we consider is

$$\text{Energy}_p(\rho) = \sum_{e \in E} \rho(e)^p, \tag{4.3}$$

This point of view allows for much more flexibility, because it can be applied to a variety of different families of objects: walks, cycles, tress, etc, and also works when the underlying network is directed or weighted. Moreover, modulus has very useful properties of

Γ-monotonicity and countable subadditivity.

Furthermore, in the special case of connecting families, by varying the parameter p, we see that $\text{Mod}_p(\Gamma(a,b))$ generalizes classical measures such as shortest path, effective resistance and min cut[51;55]. For instance, when the network is undirected and $p = 2$, $\text{Mod}_2(\Gamma(a,b))$ is exactly the effective conductance between a and b. In particular, effective conductance can be written as (see Section 2.4.1)

$$\mathcal{C}_{\text{eff}}(a) = \sum_{b \in V \setminus a} \text{Mod}_2(\Gamma(a,b)) \tag{4.4}$$

For the rest of this paper, we consider $p = 2$ due to its physical interpretation as effective conductance as well as computational advantages, for instance, in this case (4.2) is a quadratic program. Moreover, the right-hand side also makes sense on directed networks.

As mentioned above, $\mathcal{C}_{\text{eff}}(a)$ is sociocentric in the sense that it considers all walks from a to an arbitrary node in G. However, in practice, it can be prohibitive to scale sociocentric methods to very large networks. Moreover, in real-world situations it is not feasible to have access to the entire network. Rather, one can at best know local information up to a few neighborhood levels. For instance, when data is anonymized to protect privacy of network entities, identifying the sociocentric picture is impossible, e.g., sexual networks may be limited to the number of contacts of individuals.

An alternative approach is to consider measures that are adapted to egonetworks (also known as neighborhood networks). An ego network $G^a(r)$ around a node a is constructed by collecting data (nodes and edges) starting from the ego a and searching G out to a predefined order of neighborhood $r \in \{1, \cdots \epsilon(a)\}$; where $\epsilon(a)$ is the eccentricity of node a or the maximum distance from a to nodes in G.

Egonetworks are often preferred because they support more flexible data collection methods[76] and often involve less expensive computation costs. In this paper, we focus on centrality measures that are adapted to the ego-centric paradigm as substitutes for sociocentric methods, with a focus on the scalability issue. These measures are more stable[41] against network sampling and reliable (less sensitivity) with measurement errors[42]. We concentrate on unweighted (binary) networks to simplify the algebra, although, all of our methods and discussions can be easily generalized for weighted networks. Thus, we let $d(a,b)$ denote the shortest-path distance between two nodes (smallest number of hops). The neighborhood structure around an ego a is described by the shells of order k:

$$S(a,k) := \{y \in V : d(a,y) = k\},$$

and the corresponding families of walks $\Gamma(v, S(a,k))$, consisting of simple walks that begin at ego $v \in V$ and reach $S(a,k)$ for the first time. Modulus allows a quantification of the richness of the family of walks, i.e., a family with many short walks has a larger modulus than a family with fewer and longer walks. Here we consider *shell modulus* $\text{Mod}_2(v, S(a,k))$ which quantifies the capacity of walks emanating from the ego up to the shell $S(a,k)$[59] without having to account the data outside $G^a(k)$. In particular, we propose the following egocentric version of $\mathcal{C}_{\text{eff}}(a)$:

$$\mathcal{C}_{\text{shell}}(a,r) \triangleq \sum_{k=1}^{r} \text{Mod}_2(v, S(a,k)) \tag{4.5}$$

which we call *shell modulus centrality* and follows the same logic as (4.4) but only requires the egocentric network data.

4.1.1 Formula for $\mathcal{C}_{\text{shell}}(a,r)$ in undirected networks

Similar to Section 2.4.1, to find $\text{Mod}(a, S(a,r))$ in the egocentric network $G^a(r)$, we solve the equation for Kirchhoff's law of currents

$$L^a_{(r)}\mathbb{V} = \mathbb{I} \tag{4.6}$$

where $L^v_{(r)}$ is Laplacian of $G^a(r)$ and $\mathbb{I}$ is the injected external current vector with values 1 at ego and for nodes in $S(a,r)$

$$\mathbf{1}^T\mathbb{I}_S = -1 \tag{4.7}$$

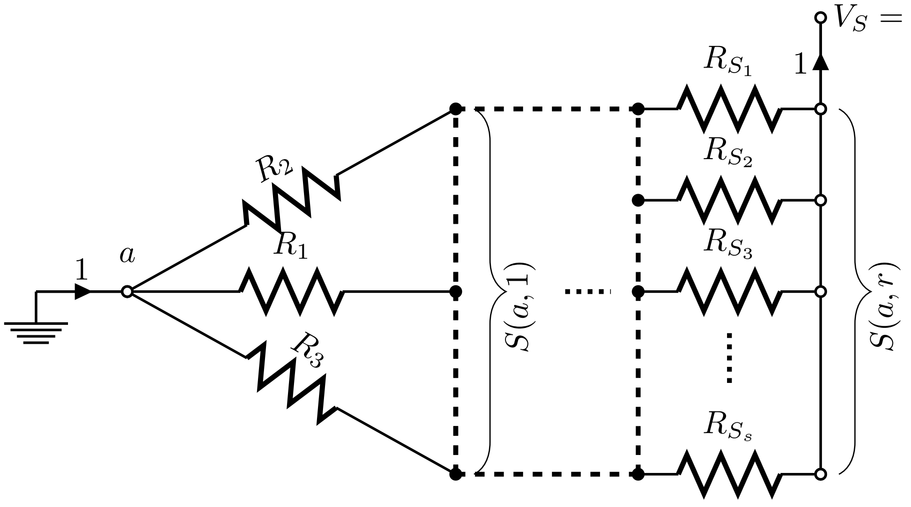


Figure 4.1: Interpreting $\mathrm{Mod}(a, S(a,r))$ as finding effective conductance between grounded node a and nodes with the same potential c in $S(a,r)$ in an electrical network problem. Solution follows from the corresponding Laplacian system.

and zero for other nodes (see Figure 4.1).

Nodes in $S(a,r)$ will have a same electric potential c, i.e., they are short circuited.

The above problem has a unique harmonic solution for $\mathbb{V}$ up to a constant, we ground the potential at ego, i.e., $\mathbb{V}_a = 0$ and find other nodes potentials by

$$\mathbb{V} = \mathcal{G}\ \mathbb{I}$$

where $\mathcal{G} = \left({}^{a}L^{v}_{(r)}\right)^{-1}$ is the reduced conductance matrix. Combining (4.6) and (4.7)

$$\begin{bmatrix} V_2 \\ \vdots \\ c \\ c \\ \vdots \\ c \end{bmatrix} = \mathcal{G} \begin{bmatrix} 0 \\ \vdots \\ \mathbb{I}_{S_1} \\ \mathbb{I}_{S_2} \\ \vdots \\ \mathbb{I}_{S_s} \end{bmatrix} \rightarrow \begin{bmatrix} V_2 \\ \vdots \\ \frac{c}{\mathbb{I}_{S_1}} \\ \frac{c}{\mathbb{I}_{S_2}} \\ \vdots \\ \frac{c}{-1-\sum_{j=S_1}^{S_s-1}\mathbb{I}_j} \end{bmatrix} = \mathcal{G} \begin{bmatrix} 0 \\ \vdots \\ 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} = \mathbf{x} \tag{4.8}$$

where $x_i = \sum_{j=S_1}^{S_s-1} \mathcal{G}_{ij}$. If $|S| = s$ and for $i \in \{S_1, \cdots, S_{s-1}\}$

$$\mathbb{I}_i = \frac{c}{x_i}$$

We can find c from last equation of (4.8):

$$\frac{c}{-1 - c \sum_{j=S_1}^{S_s-1} \frac{1}{x_i}} = x_{S_s}$$

$$c = \frac{-x_s}{1 + x_s \sum_{j=S_1}^{S_s-1} \frac{1}{x_i}}$$

and the effective resistance between a and $S(a, r)$:

$$\mathcal{R}_{a,S(a,r)} = \mathbb{V}_v - c = \frac{x_s}{1 + x_s \sum_{j=S_1}^{S_s-1} \frac{1}{x_i}}$$

and since $\mathbb{V}_a = 0$ (grounded):

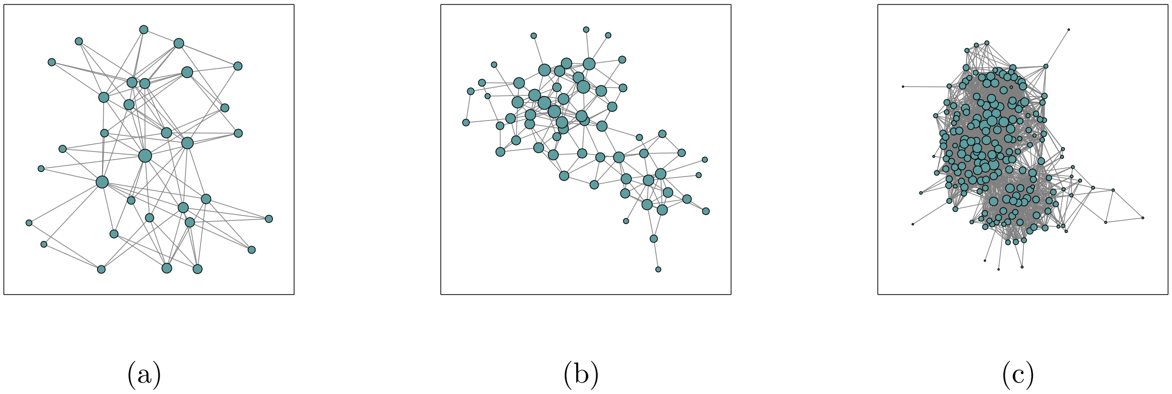
$$\mathrm{Mod}(a, S(a, r)) = \frac{1 + x_s \sum_{j=S_1}^{S_s-1} \frac{1}{x_i}}{x_s}. \tag{4.9}$$

Note that, we can ground any node other than nodes in the shell, because they have similar potentials, unless we ground all of them. This forced us to ground only the ego and thus we need one matrix inversion per ego network.

In Figure 4.2, centrality of nodes in three small networks are computed, where we consider the entire network and both $\mathcal{C}_{\text{shell}}(v, r)$ and $\mathcal{C}_{\text{eff}}(a)$ are highly correlated.

In Figure 4.2, node sizes are scaled with their $C_{\text{shell}}(v, r)$ values and the computed centralities are, as expected, highly correlated with $C_{\text{eff}}(a)$ with Spearman rank correlation 1, 0.94, and 0.99 respectively for Figures left to right meaning they are measuring a similar quantity.

For undirected networks, we can calculate both $\mathcal{C}_{\text{eff}}(a)$ and $C_{\text{shell}}(v, r)$ analytically without going through the optimization problem in (4.2) by formulas (2.10) and (4.9)



(a) (b) (c)

Figure 4.2: (a) Davis women club (b) Dolphin network (c) Jazz musicians network. Node sizes are scaled with their centrality computed by (4.5). The resultant centralities are highly correlated to (4.4) with Spearman rank correlation 1, 0.94, and 0.99 respectively for (a)-(c).

In general, (4.4) requires $|V|$ modulus computations in all of G, while (4.5) only needs r modulus computations in $G^a(r)$.

Shell modulus centrality can handle fairly large networks, e.g. 100,000 edges. The algorithm used here computes (4.2) using an active set dual method quadratic programming[63]. We have shown that it's theoretically enough to consider at most $|E|$ active constraints[50]. Violated (active) constraints are found using Dijkstra's algorithm and constraint matrix updating is done using the Cholesky decomposition.

In the following, we focus on approximating (4.5) efficiently, while incorporating most of the benefits of shell modulus in a scalable framework.

4.1.2 Bounding from above

First, we provide an upper bound that is known in the complex analysis literature as *Ahlfors estimate*[77] Chapter 4, Equations 4-6, and in the context of electrical networks goes under the name of Nash-Williams inequality[78]. Given an egonetwork $G^a(r)$, we consider the set of edges that connect a shell $S(a, k-1)$ to the next shell $S(a, k)$, for $k \in \{1, \cdots, r\}$:

$$E(a,k) := \{e = \{x,y\} \in E |\ x \in S(a,k-1),\ y \in S(a,k)\}\,.$$

We call the sets $E(a,k)$ *shell connecting sets*. Since $\text{Mod}_2(v, S(a,r))$ is a minimization problem (4.2), we get an upper bound simply by choosing an appropriate admissible density $\bar{\rho}$. Here, we pick the best admissible density that is constant for all edges in each shell connecting set. After computing the energy of this density, we obtain:

$$\text{Mod}_2(a, S(a,r)) \leq \frac{1}{\sum_{k=1}^{r} \frac{1}{|E(a,k)|}}. \tag{4.10}$$

Proof. To find an upper bound for the shell modulus $\text{Mod}_2(v, S(v,r))$, since (4.2) is a minimization problem, it is enough to pick an appropriate density $\bar{\rho}$. Here we will restrict ourselves to densities that are constant on the shell connecting sets $E(v,k)$. So consider weights x_k for $k = 1, \dots, r$ and set

$$\bar{\rho}(e) := x_k \qquad \text{if } e \in E(v,k).$$

Then we solve the following minimization problem:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \sum_{k=1}^{r} \theta_k x_k^2 \\ \text{subject to} \quad & \sum_{k=1}^{r} x_k = 1 \end{aligned} \tag{4.11}$$

where $\theta_k := |E(v,k)|$ with θ_k. By Cauchy-Schwarz inequality

$$1 \leq \left(\sum_{k=1}^{r} x_k \right)^2 = \left(\sum_{k=1}^{r} \frac{1}{\sqrt{\theta_k}} \sqrt{\theta_k} x_k \right)^2 \leq \sum_{k=1}^{r} \frac{1}{\theta_k} \sum_{k=1}^{r} \theta_k x_k^2$$

and thus the minimum in 4.11 is greater than $\left(\sum_{k=1}^{r} \frac{1}{\theta_k} \right)^{-1}$. However, when x takes the form:

$$x_k = \frac{C}{\theta_k}$$

Then the minimum is achieved for

$$C = \frac{1}{\sum_{k=1}^{r} \frac{1}{\theta_k}}.$$

□

4.1.3 Estimate for the Ahlfors upper bound

In order to study the Ahlfors upper bound, it is useful to establish the following estimate.

Claim 4.1.1. *For every finite sequence of positive numbers a_n, we have*

$$\sum_{n=1}^{N} \frac{1}{\sum_{k=1}^{n} \frac{1}{a_k}} \leq \frac{4}{3} \sum_{k=1}^{N} \frac{a_k}{k}.$$

Proof. We know that $\sum_{k=1}^{n} k = \frac{n(n+1)}{4}$. Using Cauchy-Schwarz:

$$\frac{n^2(n+1)^2}{4} = \left(\sum_{k=1}^{n} k \right)^2 = \left(\sum_{k=1}^{n} k \sqrt{a_k} \frac{1}{\sqrt{a_k}} \right)^2 \leq \sum_{k=1}^{n} k^2 a_k \sum_{k=1}^{n} \frac{1}{a_k} \tag{4.12}$$

Thus, interchanging the order of summation:

$$\sum_{n=1}^{N} \frac{1}{\sum_{k=1}^{n} \frac{1}{a_k}} \leq 4 \sum_{n=1}^{N} \frac{1}{n^2(n+1)^2} \sum_{k=1}^{n} k^2 a_k = 4 \sum_{k=1}^{N} k^2 a_k \sum_{n=k}^{N} \frac{1}{n^2(n+1)^2} \tag{4.13}$$

For every $k \geq 1$ we have:

$$\sum_{n=k}^{N} \frac{1}{n^2(n+1)^2} \leq \frac{1}{3k^3}$$

To see this considering the following sequence:

$$x_k := \frac{1}{3k^3} - \sum_{n=k}^{N} \frac{1}{n^2(n+1)^2}.$$

Then

$$x_k - x_{k+1} = \frac{1}{3k^3} - \frac{1}{3(k+1)^3} - \frac{1}{k^2(k+1)^2} = \frac{1}{3k^2(k+1)^2} > 0$$

Therefore, $x_{k+1} > 0$ implies that $x_k > 0$. So we only need to check $k = N$. But

$$x_N = \frac{1}{3N^3} - \frac{1}{N^2(N+1)^2} > 0 \qquad \forall N \in \mathbb{R}.$$

□

4.1.4 Ahlfors upper bound for Erdős-Rényi

We can provide a better estimate for Ahlfors' upper bound for Erdős-Rényi graphs in the *connected* regime:

$$p(N-1) = 2\log N.$$

Concavity of the Ahlfors bound

We can use concavity and get

$$\mathbb{E}\left(\sum_{k=1}^{r} \frac{1}{\sum_{j=1}^{k} \frac{1}{\theta_j}}\right) \leq \sum_{k=1}^{r} \frac{1}{\sum_{j=1}^{k} \frac{1}{\mathbb{E}\theta_j}}$$

So we would like to estimate $\mathbb{E}(\theta_k)$.

Computing the first two cases

- First note that θ_1 is Binomial$(N-1, p)$. So:

  $$\mathbb{E}\theta_1 = p(N-1),$$

  from the binomial distribution.

- Now, given θ_1 we must toss θ_1 variables distribute as Binomial$(N-1-\theta_1, p)$, because the ego and the first shell are now out of consideration. So

$$\mathbb{E}\left(\theta_2 \mid \theta_1\right) = \theta_1 p(N-1-\theta_1).$$

Therefore, computing the secon moment of θ_1 we get:

$$\mathbb{E}\theta_2 = \mathbb{E}(\mathbb{E}(\theta_2 \mid \theta_1)) = \mathbb{E}(\theta_1)p(N-1) - p\mathbb{E}(\theta_1^2) = p^2(1-p)(N-1)(N-2).$$

- Given θ_1 and θ_2 we must toss a certain number s of Binomial$(N-1-\theta_1, p)$ random variables, where s is the number of nodes in the second shell. However, this number s is not easy to calculate because it depends on the interaction at the previous step. For instance, if all the binomial variables in the previous step are equal to zero, then $s = 0$. But for higher values of s it becomes quite complicated.

In particular, we will have

$$\mathbb{E}\theta_1 = \log N \qquad \text{and} \qquad \mathbb{E}\theta_2 \simeq (\log N)^2.$$

Lower bound for $\mathbb{E}(\theta_k)$

First we will estimate $\mathbb{E}\theta_k$ from below. Given an ego a, Spielman sets

$$r(a) := \max\left\{r : |B(r,a)| \leq \frac{N}{12\log N}\right\}$$

and then shows that for $k \leq r(a)$,

$$\mathbb{P}\left[|S(a,k+1)| \leq \frac{1}{5}\log N|S(a,k)|\right] \leq N^{-1.2|S(a,k)|}.$$

He first finds that

$$\mathbb{E}\left[|S(a,k+1)| \mid G^a(k)\right] \geq \frac{5}{3}|S(a,k)|\log N, \tag{4.14}$$

and then applies the theory of Chernoff bounds. Note that by simply taking the expectation in (4.14) we get

$$\mathbb{E}|S(a,k+1)| \geq \frac{5}{3}(\log N)\mathbb{E}|S(a,k)|.$$

This gives geometric growth for $k \leq r(a)$:

$$\mathbb{E}|S(a,k)| \geq (\log N)^k. \tag{4.15}$$

In our case, since every $c \notin B(a,k)$ must toss $|S(a,k)|$ biased coins, we get

$$\mathbb{E}\left[\theta_{k+1} \mid G^a(k)\right] = |S(a,k)|p(N - |B(a,k)|) \geq \frac{11}{12}|S(a,k)|pN = \frac{11}{6}(\log N)|S(a,k)|.$$

Again we can take expectations and get

$$\mathbb{E}\theta_{k+1} \geq \frac{11}{6}(\log N)\mathbb{E}|S(a,k)|.$$

Using (4.15), we get

$$\mathbb{E}\theta_k \geq (\log N)^k.$$

Upper bound for $\mathbb{E}\theta_k$

To get an upper bound we can compare the growth in the Erdős-Rényi graph with the growth for a Galton-Watson branching process with offspring distribution $X = \text{Binomial}(N-1,p)$. This will be larger because there are no collisions and we always toss the maximum number of coins. If Z_k is the population at time k, then

$$\mathbb{E}Z_k = \mu^k$$

where $\mu = \mathbb{E}X = p(N-1) = 2\log(N)$. So we get that

$$\mathbb{E}\theta_k \leq (2\log N)^k.$$

Upper bound for the Ahlfors estimate

We can apply this to our estimate of the average Ahlfors upper bound and get that:

$$
\begin{aligned}
\mathbb{E}\left(\sum_{k=1}^{r}\frac{1}{\sum_{j=1}^{k}\frac{1}{\theta_j}}\right) &\le \sum_{k=1}^{r}\frac{1}{\sum_{j=1}^{k}\frac{1}{\mathbb{E}\theta_j}}\\
&\le \sum_{k=1}^{r}\frac{1}{\sum_{j=1}^{k}\frac{1}{(2\log N)^j}}\\
&= (2\log N - 1)\sum_{k=1}^{r}\frac{1}{1-(2\log N)^{-k}}\\
&= (2\log N - 1)\sum_{k=1}^{r}\frac{(2\log N)^k}{(2\log N)^k - 1}\\
&= (2\log N - 1)\sum_{k=1}^{r}\left[1+\frac{1}{(2\log N)^k - 1}\right]\\
&\simeq (2\log N - 1)\left[r+\sum_{k=1}^{r}\frac{1}{(2\log N)^k}\right]\\
&= (2\log N - 1)\left[r+\frac{1}{2\log N}\frac{1-\left(\frac{1}{2\log N}\right)^r}{1-\frac{1}{2\log N}}\right]\\
&= (2\log N - 1)\left[r+1-\frac{1}{(2\log N)^r((2\log N)-1)}\right]\\
&\simeq (2\log N - 1)(r+1)
\end{aligned}
$$

4.1.5 Bounding shell modulus from below

To provide a lower bound for shell modulus, we focus on geodesic paths (shortest walks). These are usually the most important pathways of influence between the ego and other nodes. Classical measures of centrality, such as closeness centrality and betweenness centrality, are based uniquely on shortest paths[40].

When collecting the egocentric data around an ego a, one can take care to avoid forming cycles, and the resulting egonetwork becomes a tree. So assuming $T^a(r)$ is a tree contained in $G^a(r)$, we can use Γ-monotonicity to get a lower bound. Moreover, if we write $\mathrm{Mod}_2(T^a(r))$ for the shell modulus of all walks in $T^a(r)$ starting at the root a and reaching depth-level r.

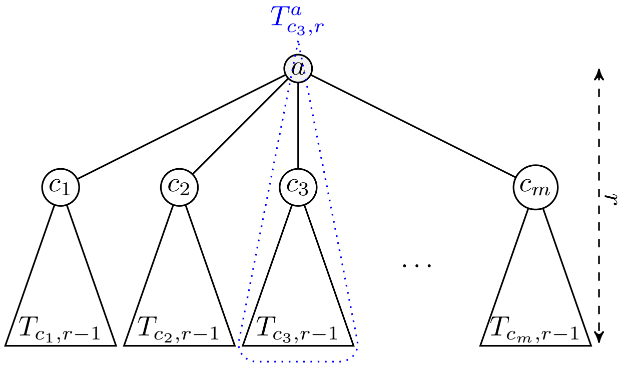


Figure 4.3: The tree T^a_r and its subtrees. Each child c_i of a can induce two subtrees–if it has descentants until depth $r-1$. $T^a_{c_i,r}$ (outlined with a dashed line for $i = 3$ in the figure) is the subtree rooted at v formed by removing all other children and their descendants from T^a_r. $T_{c_i,r-1}$ is the subtree rooted at c_i formed by removing a from $T^a_{c_i,r}$.

Let T_a be a rooted shortest tree at a with vertex set V, and edge set E. Every density $\rho : E \to [0, \infty)$ gives a weighted distance on the tree defined by

$$d_\rho(x, y) = \sum_{e \in \gamma(x,y)} \rho(e)$$

We define the set of admissible densities $\mathrm{Adm}(T^a_k)$, for walks starting from root a (ego) to leaves at depth k, denoted by l_k:

$$\mathrm{Adm}(T^a_k) := \{\rho : E \to [0, \infty) : \ell_\rho(a, l_k) \geq 1\}.$$

with modulus

$$\mathrm{Mod}_2(T^a_k) := \inf_{\rho \in \mathrm{Adm}(T_{a,k})} \sum_{e \in E} \rho(e)^2$$

Assuming a has at least one child, let $C(a) := \{c_1, c_2, ...\} \subseteq V$ be the children. Each child c induces two rooted subtrees (Figure 4.3). Let T^a_c represent the subtree (still rooted at a) formed from T_a by pruning all of as children other than c along with their descendants, and let T_c represent the subtree (now rooted at c) formed by removing a from T^a_c.

The following lemma is an immediate consequence of the *parallel rule* of modulus, i.e., given two families Γ_1 and Γ_2, where for every $e \in E$ and $\gamma_1 \in \Gamma_1$ and $\gamma_2 \in \Gamma_2$ we have

$\mathcal{N}(\gamma_1, e)\mathcal{N}(\gamma_2, e) = 0$. Thus $\mathrm{Mod}_2(\Gamma_1 \cup \Gamma_2) = \mathrm{Mod}_2(\Gamma_1) + \mathrm{Mod}_2(\Gamma_2)$.

Lemma 4.1.2. *The modulus of T_k^a is related to the moduli of the $T_{c_i,k}^a$ as follows.*

$$\mathrm{Mod}_2(T_k^a) = \sum_{i=1}^{m} \mathrm{Mod}_2(T_{c_i,k}^a).$$

By Lemma 4.1.2, we may restrict ourselves to the case that a has a single child c. In this case, *serial rule* for modulus allows us to reduce the problem to finding the modulus of $T_{c,k-1}$. This is explained in the following lemma.

Lemma 4.1.3. *The modulus of $T_{c,k}^a$ is related to the modulus of $T_{c,k-1}$ as follows.*

$$\mathrm{Mod}_2(T_{c,k}^a) = \frac{\mathrm{Mod}_2(T_{c,k-1})}{1 + \mathrm{Mod}_2(T_{c,k-1})} \tag{4.16}$$

Proof. If c is a leaf of T_k^a , then $\rho(a, c) = 1$ is the minimizer for the modulus. Otherwise, by considering the density, $\rho(a, c)$, on the edge from a to c, the optimization effectively decouples. In order for ρ to be admissible, it is necessary that $d_\rho(c, l) \geq 1 - \rho(a, c)$ for every leaf l_{k-1} of $T_{c,k-1}$ at depth $k - 1$. For $0 \leq \ell \leq 1$, define the parameterized set of admissible densities, for every leaf l_{k-1}

$$\mathrm{Adm}(T_{c,k-1}; \ell) := \{\rho : E \to [0, \infty) : d(c, l_{k-1}) \leq \ell\}$$

and the parameterized modulus problem

$$\mathrm{Mod}'_p(T_{c,k-1}; \ell) = \inf_{\rho \in \mathrm{Adm}'(T_{c,k-1};\ell)} \sum_{e \in E(T_c)} \rho(e)^2$$

where $E(T_{c,k-1})$ represents the set of edges in the subtree $T_{c,k-1}$. It is straightforward to verify that

$$\mathrm{Mod}'_2(T_{c,k-1}; \ell) = \ell^2 \, \mathrm{Mod}_2(T_c)$$

and, thus

$$\begin{aligned}\operatorname{Mod}_2(T^a_{c,k}) &= \inf_{0\le\rho(a,c)\le 1}\{\rho(a,c)^2+\operatorname{Mod}_2'(T_{c,k-1}:1-\rho(a,c))\}\\ &= \inf_{0\le\rho(a,c)\le 1}\{\rho(a,c)^2+(1-\rho(k,c))^2\operatorname{Mod}_2(T_{c,k-1})\}\end{aligned} \tag{4.17}$$

The infimum, given by (4.16), is attained when

$$\rho(a,c)=\frac{\operatorname{Mod}_2(T_{c,k-1})}{1+\operatorname{Mod}_2(T_{c,k-1})}$$

□

Lemmas 4.1.2 and 4.1.3 combined prove the following theorem.

Theorem 4.1.4. *The modulus* $\operatorname{Mod}_2(T^a(k))$ *can be found by the formula*

$$\operatorname{Mod}_2(T^a_k)=\sum_{c\in C(a)}\frac{\operatorname{Mod}_2(T_{c,k-1})}{1+\operatorname{Mod}_2(T_{c,k-1})} \tag{4.18}$$

Equation (4.18) computes $\operatorname{Mod}_2(T^a(k))$ recursively. For each leaf node l_k, set $\operatorname{Mod}_2(T_{l_k,0})=\infty$. Then (4.18) will propagate the modulus to the ego. For example, to compute $\operatorname{Mod}_2(T_{\mathrm{a},2})$ in the graph in Figure 4.4(b), we start by assigning ∞ for modulus of the leaves e and f. Then, by (4.18), each contributes 1 to node b, and $\operatorname{Mod}_2(T_{\mathrm{b},1})=2$. Thus $\operatorname{Mod}_2(T^{\mathrm{a}}(2))=\frac{\operatorname{Mod}_2(T_{\mathrm{b},1})}{1+\operatorname{Mod}_2(T_{b,1})}=\frac{2}{3}$.

4.2 General degree

From the previous section, Ahlfors' upper bound (4.10) considers all edges in the shell connecting sets even if they are not on the shortest paths, such as edge $a-d$ in Figure 4.4(a). On the other hand, when using the ego-tree approximation, we inevitably lose valuable information hidden in the edges that where discarded. For example in Figure 4.4(b-c), to form a tree we need to solve the child custody problem between parents b and c and child f. In

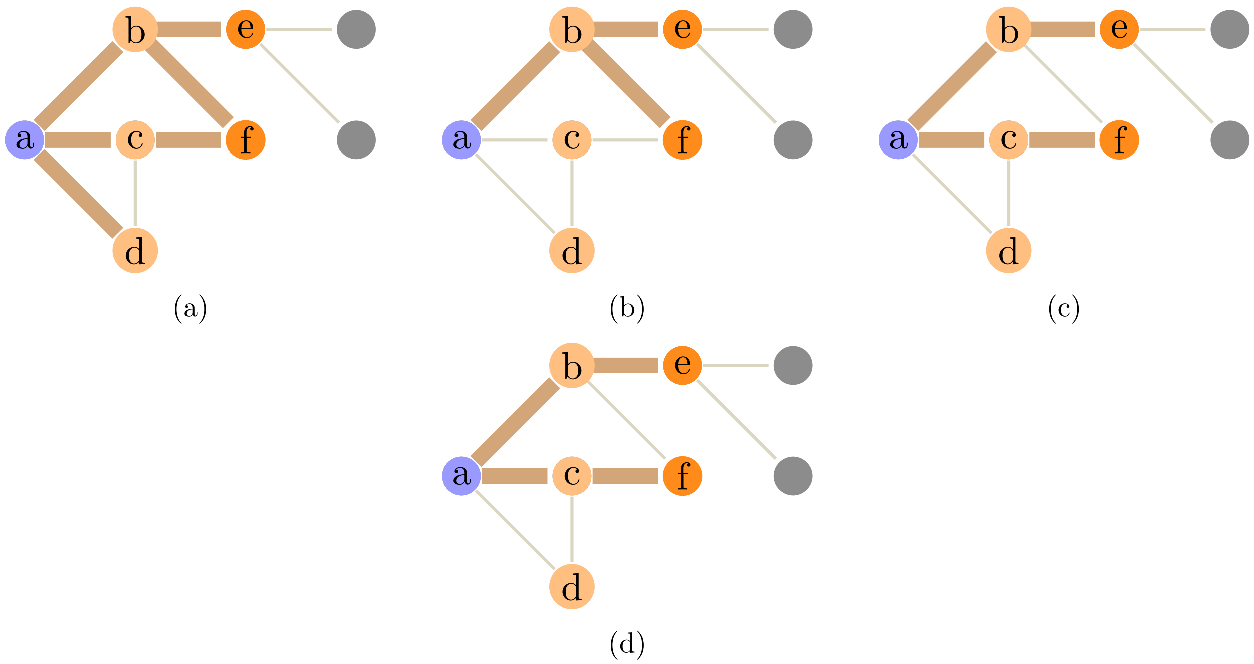


Figure 4.4: (a) To compute the upper bound in (4.10), for ego a and depth $k = 2$, edge $\{a, c\}$ has the same role as edge $\{a, d\}$. (b) and (c) give different ways to obtain T_2^a. (d) shows the edges considered in general degree.

particular, the lower bound calculation will discard at least one edge. Moreover, this leads to multiple possible lower bounds, e.g., $\mathrm{Mod}_2(T_{a,r}) = \frac{2}{3}$ in Figure 4.4(b) and $\mathrm{Mod}_2(T_{a,r}) = 1$ for Figure 4.4(d).

As a compromise between the Ahlfors upper bound and the tree modulus lower bound, we propose a measure we call *general degree*. Fix a depth $i = 1, 2, 3, \ldots, r$ and consider a tree rooted at the ego a, whose leaves are all contained in the shell $S(a, i)$, and such that the geodesics from the root to $S(a, i)$ take exactly i hops. Let $H(a, i) = (V_i, E_i)$ be the union of all such trees found by breadth first search. For instance, in Figure 4.4(d) we show $H(\mathrm{a}, 2)$ in that case. Note that we discarded nodes that are not on the geodesic paths from a to $S(a, 2)$.

Since, in general, we cannot use the recursion (4.18) on $H(a, r)$, we instead compute the upper bound (4.10). Namely, we consider the shell connecting sets $E_i(a, k)$ for $H(a, i)$ and

Algorithm 2 Algorithm for computing summands in (4.19).

```
D ← set of all descendants for each ancestor
r ← neighborhood order
k ← 1
for nodes in $\{S^r(a,k), k \leq r\}$ do
    Update D with nodes as new descendants
    Removing ancestors that do not have any descendants in nodes
    $k \leftarrow k + 1$
end for
return harmonic means of number of ancestral relations in each k
```

define general degree to be the following expression:

$$\mathrm{gDeg}(a) := \sum_{i=1}^{r} \frac{1}{\sum_{k=1}^{i} \frac{1}{|E_i(a,k)|}} \tag{4.19}$$

Observe that the first summand of 4.19 is the ordinary degree of the ego and thus our formula acts as a generalization of degree which takes into account information about the shells around the ego. For example, we have $E_1(a,1) = 3$, $E_2(a,1) = 2$, $E_2(a,2) = 3$ in Figure 4.4(d). For $r = 2$, $\mathrm{gDeg}(a) = 3 + \frac{1}{\frac{1}{2}+\frac{1}{3}} = 3 + 6/5 = 4.2$. For small depths r the computation can be done by hand with keeping track of ancestral relations from the ego to nodes in each shells resulting in $\mathcal{O}(rn_a)$ complexity for an ego network $G^a(r)$ with n_a nodes.

4.2.1 Comparisons of shell modulus approximations and an algorithm for general degree

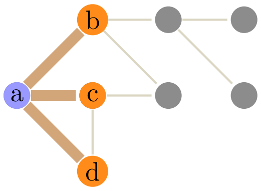
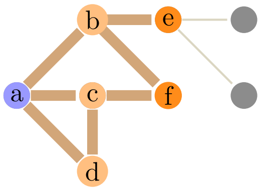
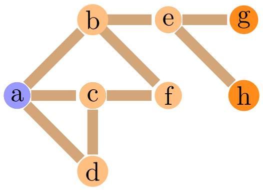
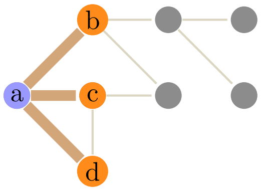
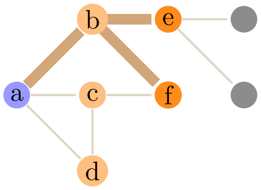
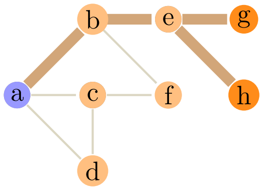
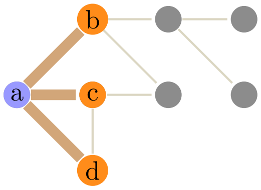
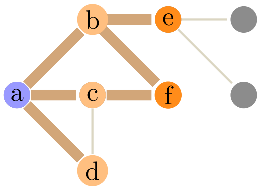
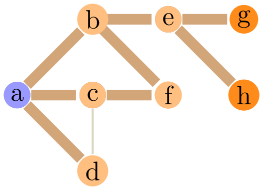
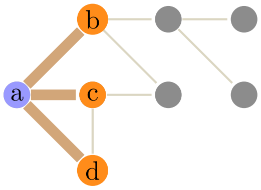
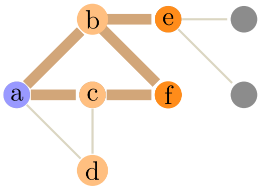
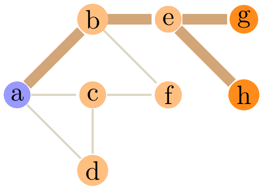
We illustrate the differences between the proposed method with (4.5), (4.10), and (4.18) in Table 4.1 for a small example egocentric network.

Genral degree, behaves similar to degree and no normalization is needed which is critical when comparing centrality of different egos, when there is no information about connections between their ego-networks.

We can compute the summands in (4.19) with Algorithm 2.

In short, we keep track of ancestral relations from the ego to nodes in each shells, and

Table 4.1: Examples for Shell modulus, bounds and general degree.

Quantity	$i = 1$	$i = 2$	$i = 3$	total
$\mathrm{Mod}(v, S_i)$	 3	 1.26	 0.44	4.71
Lowerbound	 3	 0.66	 0.4	4.06
Upperbpund	 3	 1.5	 0.85	5.35
General Degree	 3	 1.2	 0.4	4.6

discard nodes those that do not have any descendants in shell r; leading to required information about $H(a, r)$ and thus we can find summands in (4.19). The overall time complexity of calculating (4.19) is due to the graph search in step 4 of Algorithm 2 and keeping the informationc of ancestral relationships, i.e, for an ego network $G^a(r)$ size n_a, algorithm performance is in $\mathcal{O}(rn_a)$.

We illustrate the performance of general degree compared to the Ahlfors upper bound and the Tree modulus lower bound for conventional random network models such as Erdős-Rényi networks, scale-free (Barabasi-Albert model[8]), Spatial (geometric model in the unit square[9]), and small world (Watts-Strogatz model[11]). Figure 4.5 shows that general degree gives a better approximation for $\mathcal{C}_{\text{shell}}(a, r)$ than the Ahlfors and Tree modulus estimates.

4.3 Applications and results

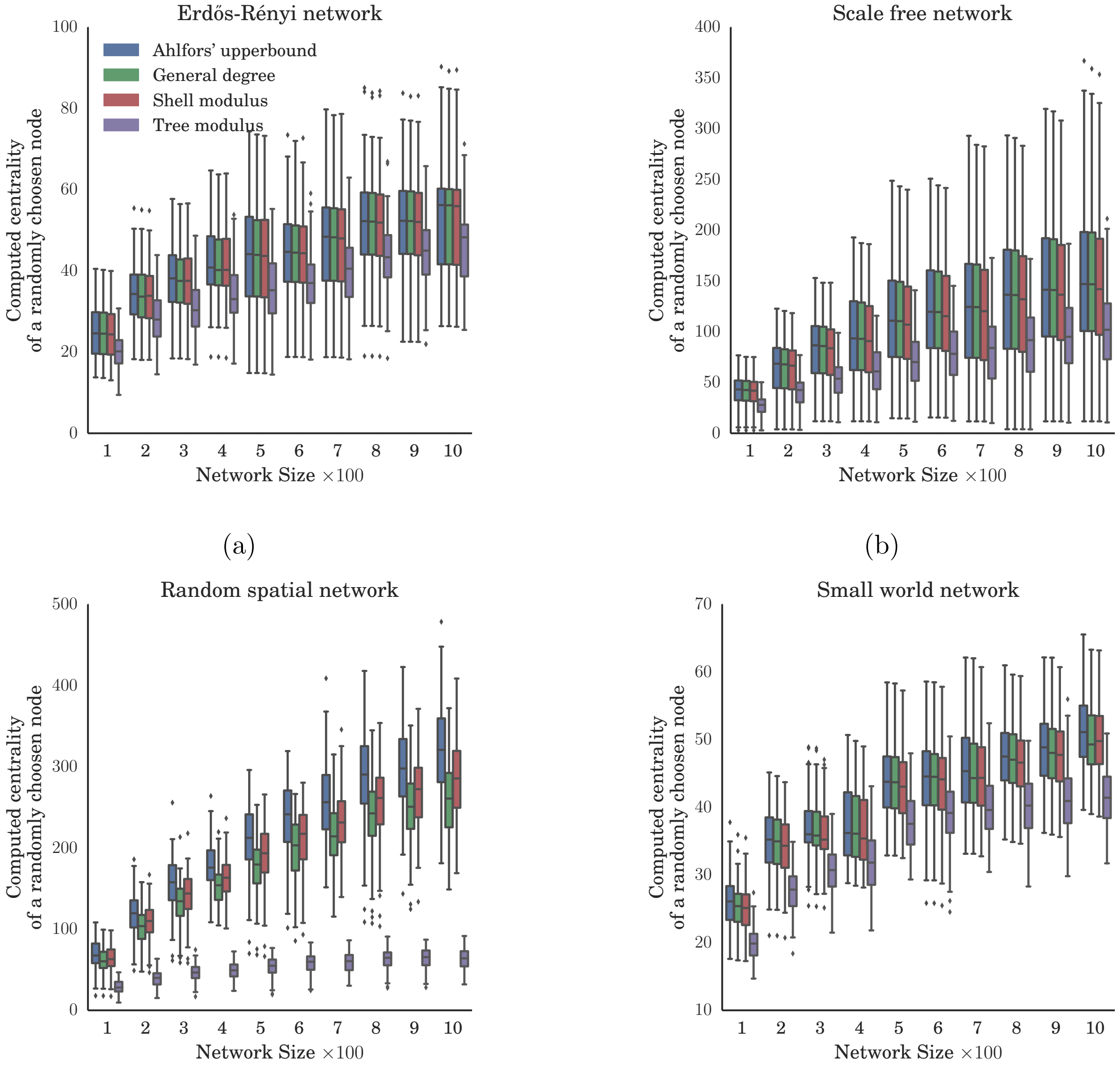
Computing node centrality in networks has numerous practical applications, for example, finding influential nodes in immunization strategies. We evaluate the proposed measures with comparison to other existing popular centralities. Although, comparing egocentric measures to sociocentric ones is in favor of the latter, we can use the results to evaluate their performance.

Because $\mathcal{C}_{\text{eff}}$ is a widely accepted measure and efficient algorithms are available for medare size undirected networks[7], we focus on benchmarking our ego-centric measures with this sociocentric counterpart (4.4) in the subsequent discussions.

In this section, we will consider networks in Table 4.2.

4.3.1 Effects of neighborhood order r on general degree

We examine the correlation of the computed general degree when considering increasing order of neighborhood cutoff r in (4.19) with the sociocentric data (entire network). Our studies show that for most of the networks $cutoff = 3$ can gives over %90 correlation with respect to having the entire data in the measure. In Figure 4.6, we illustrate two examples



(a)

(b)

(c)

(d)

Figure 4.5: Comparing the value of the Ahlfors upper bound, Tree modulus lower bound, General degree, and Shell modulus in randomly generated network models (a) Erdős-Rényi networks with $p = 2\log n/n$, (b) Scale free network by Barabasi and Albert model[8] with 6 edges preferential attachment. (c) Spatial network (random geometric network[9]) with distance threshold value $r = \sqrt{2\log n/n}$ and small world network by Watts-Strogatz model with initial degree of $2\log n$ and rewiring probability 0.3. General degree is providing a fair estimate of shell modulus in these networks.

Table 4.2: Network attributes.

Network	$\lvert V\rvert$	$\lvert E\rvert$
network of 40 homosexual men[10]	40	41
Bottlenose dolphins social network[79]	62	59
Jazz musician collaboration[14]	198	2,742
Davis southern club women[80]	32	89
Power grid of western united states[11]	4,941	6,594
Users interaction network of Pretty Good Privacy (PGP)[12]	10,680	24,316
Facebook friendship network of Princeton University[13]	13,081	88,266

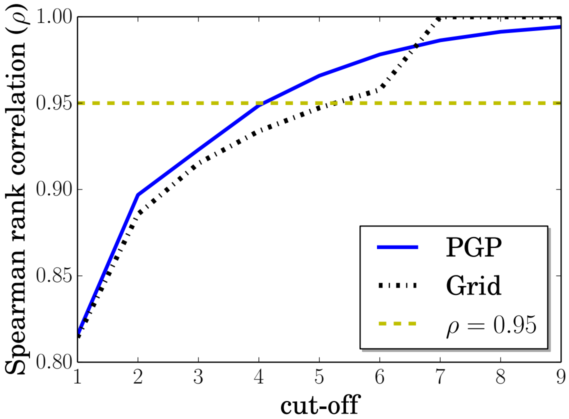


Figure 4.6: Correlation of values of gDeg for networks PGP and power grid computed in different cutoffs.

of PGP network and US power grid.

4.3.2 Ranking of nodes

In the seminal paper of Stephenson and Zelen[73], authors study a network of 40 homosexual men[10] (Figure 4.7(top)) and they show the advantages of information centrality $\mathcal{C}_{\text{eff}}$ and they suggest ranking the nodes based on this measure represents the node overall structural importance in the network. The resultant ranking is useful in detecting individuls that transfer HIV virus easily. We redo the experiment and compute the centrality of nodes with $\mathcal{C}_{\text{eff}}$ by degree and general degree in Table 4.3.

Top three central nodes are similar in both $\mathcal{C}_{\text{eff}}$ and general degree. General degree distinguishes between importance of peripheral nodes (with degree 1) such as 14 and 15 that are connected to the most central node 16. Moreover, lowest central peripheral node 35 in

Table 4.3: Ranking by different centrality measures for network of 40 homosexual men[10]

Rank	$\mathcal{C}_{\text{eff}}$	Degree	gDegree
1	16 (0.0104)	16 (8)	16 (14.096)
2	22 (0.0097)	5 (5)	26 (10.935)
3	26 (0.0096)	26 (5)	22 (9.102)
4	20 (0.0089)	22 (4)	5 (8.352)
5	11 (0.0087)	8 (3)	11 (8.350)
6	28 (0.0087)	11 (3)	28 (8.180)
7	19 (0.0083)	20 (3)	20 (7.808)
8	31 (0.0077)	28 (3)	31 (7.579)
9	14 (0.0075)	31 (3)	8 (6.360)
10	12 (0.0074)	32 (3)	19 (6.252)
11	15 (0.0074)	34 (3)	32 (6.158)
12	17 (0.0074)	38 (3)	38 (5.991)
13	21 (0.0074)	2 (2)	34 (5.323)
14	38 (0.0072)	9 (2)	14 (5.184)
15	23 (0.0072)	14 (2)	29 (5.064)
16	25 (0.0071)	19 (2)	33 (4.823)
17	27 (0.0070)	23 (2)	36 (4.766)
18	5 (0.0070)	29 (2)	23 (4.749)
19	8 (0.0068)	33 (2)	2 (4.717)
20	18 (0.0066)	36 (2)	9 (4.587)
21	29 (0.0066)	1 (1)	12 (4.234)
22	32 (0.0062)	3 (1)	15 (4.234)
23	36 (0.0060)	4 (1)	17 (4.234)
24	13 (0.0058)	6 (1)	21 (4.234)
25	39 (0.0057)	7 (1)	18 (4.078)
26	40 (0.0057)	10 (1)	27 (4.048)
27	24 (0.0056)	12 (1)	3 (3.863)
28	2 (0.0056)	13 (1)	4 (3.863)
29	3 (0.0055)	15 (1)	6 (3.863)
30	4 (0.0055)	17 (1)	25 (3.826)
31	6 (0.0055)	18 (1)	7 (3.732)
32	9 (0.0054)	21 (1)	30 (3.565)
33	7 (0.0054)	24 (1)	39 (3.559)
34	34 (0.0054)	25 (1)	40 (3.559)
35	33 (0.0054)	27 (1)	13 (3.478)
36	30 (0.0052)	30 (1)	1 (3.416)
37	37 (0.0049)	35 (1)	35 (3.415)
38	1 (0.0046)	37 (1)	37 (3.388)
39	10 (0.0045)	39 (1)	10 (3.373)
40	35 (0.0044)	40 (1)	24 (3.246)

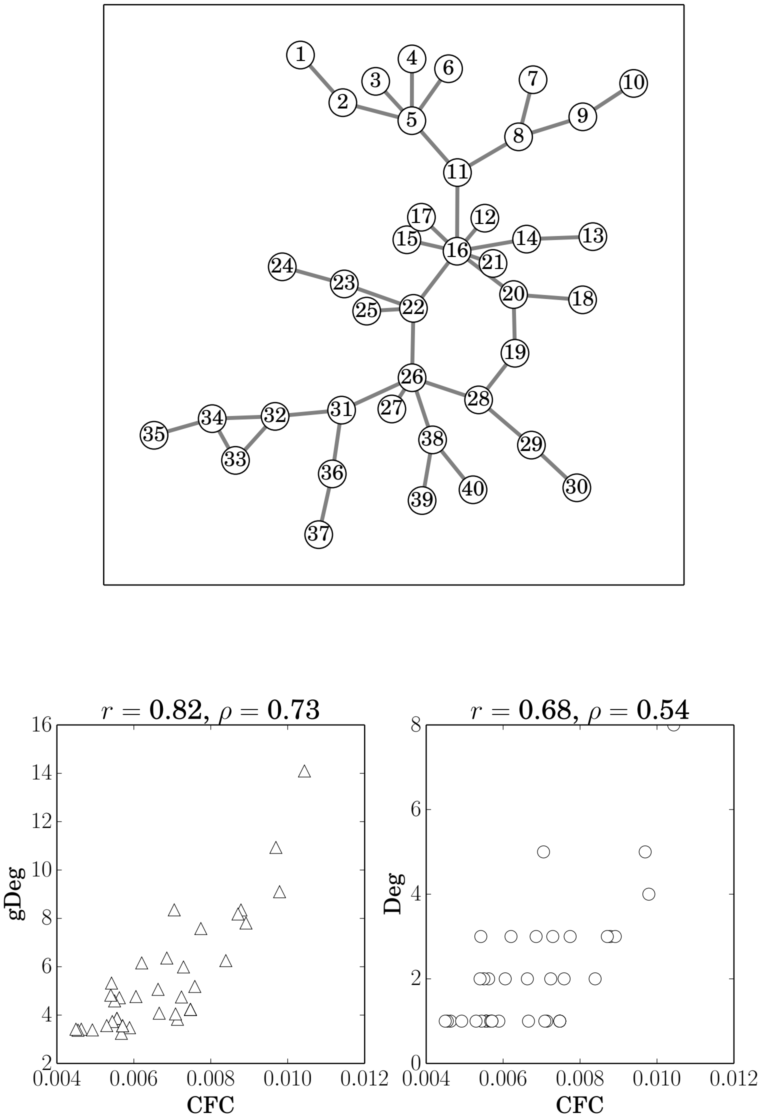


Figure 4.7: (Top) Network of 40 homosexual men[10] (Bottom) Pearson r and Spearman ρ correlation of general degree with C_{eff} (c) Pearson r and Spearman ρ correlation of degree with C_{eff}.

$\mathcal{C}_{\text{eff}}$ is also a low central node in general degree. The general degree shows a close correlation to $\mathcal{C}_{\text{eff}}$ compared to degree 4.7(b). Although, degree cannot distinguish between nodes with the same degree while they are different in the other two measures.

4.3.3 Application in immunization strategies

Targeted immunizations in computer networks and heterogenious populations can greatly impact the overall outcome of spreading processes[81–83]. Mitigating an epidemic with random immunization of nodes, requires vaccinating over %80 of the population and thus identifying a good set of target nodes has attracted much attention[84;85].

However, most of the methods for finding proper sets of nodes for immunization requires global knowledge about the network, making it impossible to use in practical situations. Therefore, scientists prefer algorithms that are agnostic to global structure of the network, for example *acquaintance immunization* chooses random neighbors of randomly picked nodes[86]. In what follows, we illustrate the immunization performance of general degree when $r = 3$, i.e., knowledge of neighbors together with neighbors of neighbors, compared to other popular methods, such as acquaintance, effective conductance, and betweenness and eigenvector centrality.

We consider the well-known spreading model susceptible, infected, recovered (SIR) that can represent infectious processes that are not reversible and susceptible nodes in the network can become infected I (proportional to infectious severity β rate and the number of infected neighbors) and eventually rest in R state (immune) after a recovery period $\frac{1}{\delta}$ days, i.e, S→I (see Figure 4.8). We assume a constant $\delta = 0.1$, i.e., nodes stay in I state in average for 10 days. To model widespread diseases such as Flu that caused by close contacts, the infectious rate β is chosen to have reproduction number $R_0 \sim \frac{\beta}{\delta}\langle k \rangle = 3$, where $\langle k \rangle$ is the avergae degree of the network[85].

We investigate the vaccination strategies that choose different fractions of population to immunize based and after updating the contact networks with the immunized nodes, we asses the performance of different strategies. In our experiments, all nodes are initially

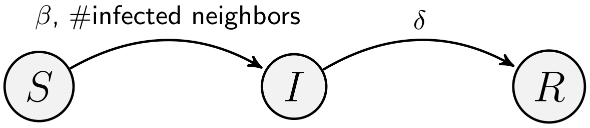


Figure 4.8: Schematic of the SIR model.

susceptible and the infectious process starts with a randomly chosen patient zero. The algorithm performances are monitored by measuring the epidemic final size, i.e., number of nodes in R state after there is no more I state nodes.

We simulate the process 2000 times to get more insights into the undergoing spreading nature in the newly obtained contact networks with different immunization strategies. The simulations are done with GEMFsim, that employs event-based exact stochastic simulation[87]. We test the significance of comparisons of the obtained results by the nonparametric Mann-Whitney test[88].

In Figure 4.9, we compare immunization performance of effective conductance, acquaintance, and betweenness and eigenvector centrality to general degree with $r = 3$.

In addition to US power grid and PGP networks, we consider the friendship network for Princeton University and University of North Carolina at Chapel Hill (UNC) extracted from Facebook[13]. To assume potential physical networks, Salathe *et. al.*[85] suggests only interactions of individuals in the same dormitory or if they are in the same year and same major. This makes the networks extremely modular and poses a big challenge for centrality measures that emphasize on closeness of the nodes to others. As it is shown in Figure 4.9, up to %15 fraction of immunization betweenness centrality measure is delivering better choice of immunization, but with considering more immunized people our egocentric measure statrs to outperform it.

Effective conductance and betweenness centrality performs better than general degree in small immunization coverages. One explanation is these centrality measures are computed for the initial networks and with removing nodes, networks are changing and the central nodes will differ consequently. Therefore, with increasing immunization coverage, General degree performs better compared to other methods, more similar to effective conductance (as expected). General degree is performing better than both eigenvector centrality and acquaintance immunization. The latter can be explained because it is considering less information than general degree.

4.3.4 Behavior of shell modulus estimates when $n, r \to \infty$

Modulus on the complete graph

Verifying that a metric ρ is extremal for p-modulus can be done using Beurling's criterion (proof in[52]).

Theorem 4.3.1 (Beurling's Criterion for Extremality)**.** *Let G be a simple graph, Γ a family of walks on G, and $1 < p < \infty$. Then, a density $\rho \in \mathrm{Adm}(\Gamma)$ is extremal for $\mathrm{Mod}_p(\Gamma)$, if there is a subfamily $\tilde{\Gamma} \subset \Gamma$ with $\ell_\rho(\gamma) = 1$ for all $\gamma \in \tilde{\Gamma}$, such that for all $h \in \mathbb{R}^E$:*

$$\textstyle\sum_{e\in E} \mathcal{N}(\gamma, e)h(e) \geq 0, \quad \textit{for all } \gamma \in \tilde{\Gamma} \quad \Longrightarrow \quad \displaystyle\sum_{e\in E} h(e)\rho^{p-1}(e) \geq 0. \tag{4.20}$$

The *complete graph* K_N is a simple graph on N nodes, where every node is connected to each other, see Figure 4.13.

Figure 4.14 depicts the extremal density ρ^* for $\Gamma(a, b)$ in K_N.

In formulas, $\rho^*(a, x) = 1/2 = \rho^*(b, x)$ for every $x \neq a, b$, and $\rho^*(a, b) = 1$, otherwise ρ^* is zero. To verify Beurling's criterion, consider the subfamily $\tilde{\Gamma}$ of simple paths consisting of $a\ b$ and $a\ x\ b$ for any $x \neq a, b$. We get that

$$\mathrm{Mod}_p(\Gamma(a, b)) = 1 + 2(N - 2)\frac{1}{2^p} \qquad \text{and} \qquad \mathrm{Mod}_2(\Gamma(a, b) = \frac{N}{2}.$$

Take n complete graphs $K_1, \ldots, K_n$.

Constant sizes For $j = 1, \ldots, n$, assume that $|V(K_j)| = N$, and pick a pair of distinct nodes $x_{j-1}, y_j \in V(K_j)$. Then, for $j = 1, \ldots, n - 1$, glue $y_j \in V(K_j)$ to $x_j \in V(K_{j+1})$. We denote the resulting graph by $G(N, n)$.

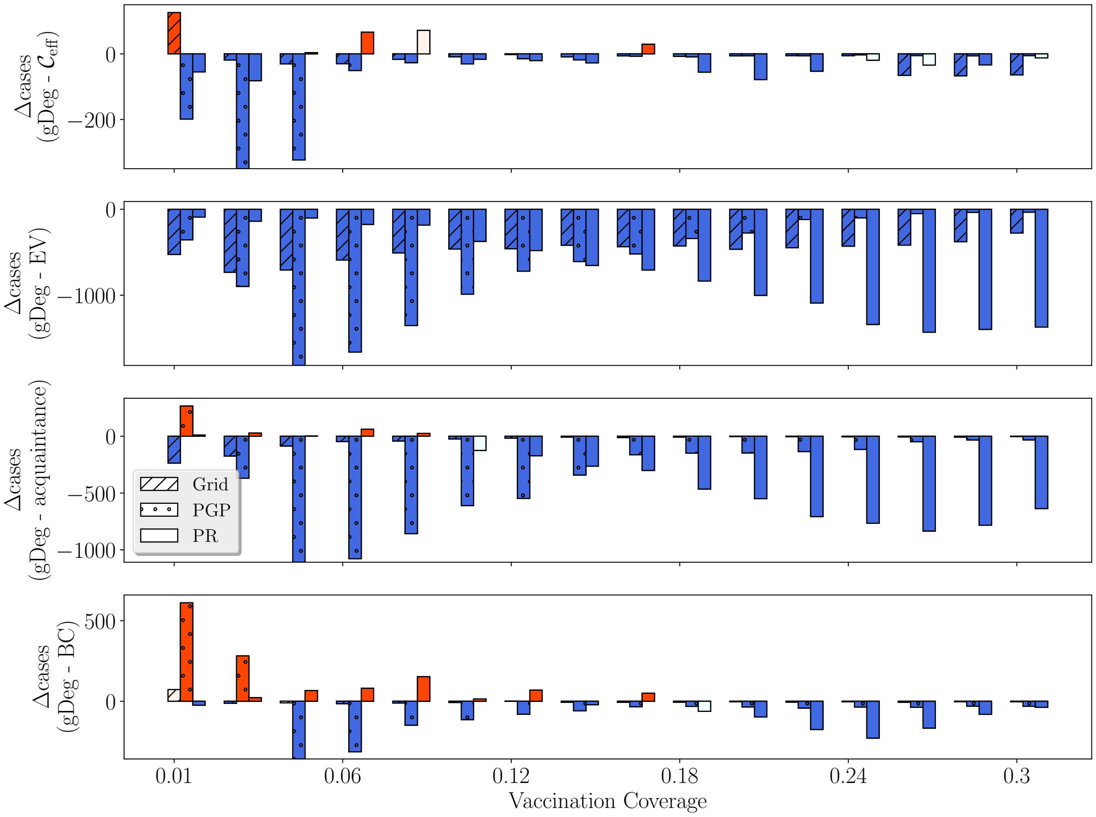


Figure 4.9: Comparing different immunization strategies with effective conductance, acquaintance, eigenvector centrality, and betweenness centrality with general degree ($r = 3$). The immunization coverage varies from %1 to %30 of the highest central nodes. Bars show the difference of final size of epidemic outbreak. Negative differences shows general degree performs better in the immunization compared to the other policy. By increasing the coverage, general degree outperforms other methods. Results are inferred by 2000 simulations of SIR epidemic model and statistically nonsignificant results are shown by shaded bars. Empirical networks are US power grid (Grid)[11], PGP network (PGP)[12], Facebook friendship network of Princeton university (PR)[13]. Statistically insignificant differences are shown by shaded colors.

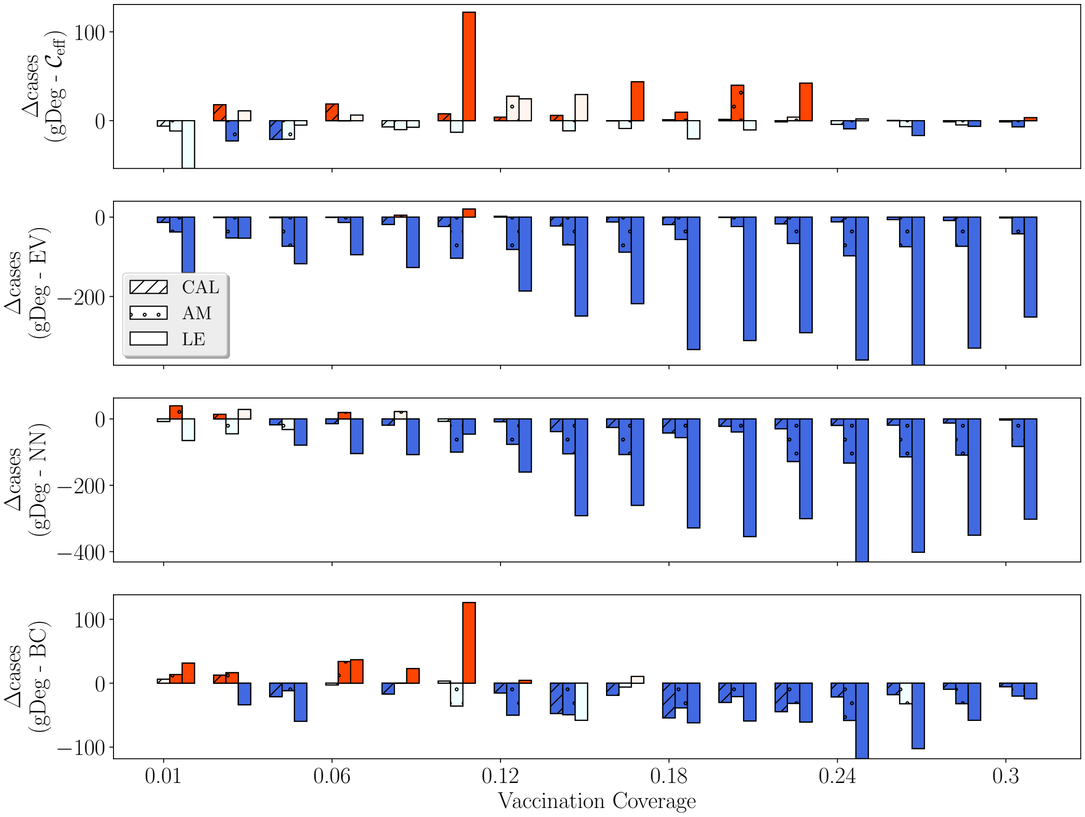


Figure 4.10: Comparing different immunization strategies with effective conductance, acquaintance, eigenvector centrality, and betweenness centrality with general degree ($r = 3$). The immunization coverage varies from %1 to %30 of the highest central nodes. Bars show the difference of final size of epidemic outbreak. Negative differences shows general degree performs better in the immunization compared to the other policy. By increasing the coverage, general degree outperforms other methods. Results are inferred by 2000 simulations of SIR epidemic model and statistically nonsignificant results are shown by shaded bars. Facebook friendship network of UC Berkeley (CAL), Amherst (AM) and Lehigh (LE). Statistically insignificant differences are shown by shaded colors.

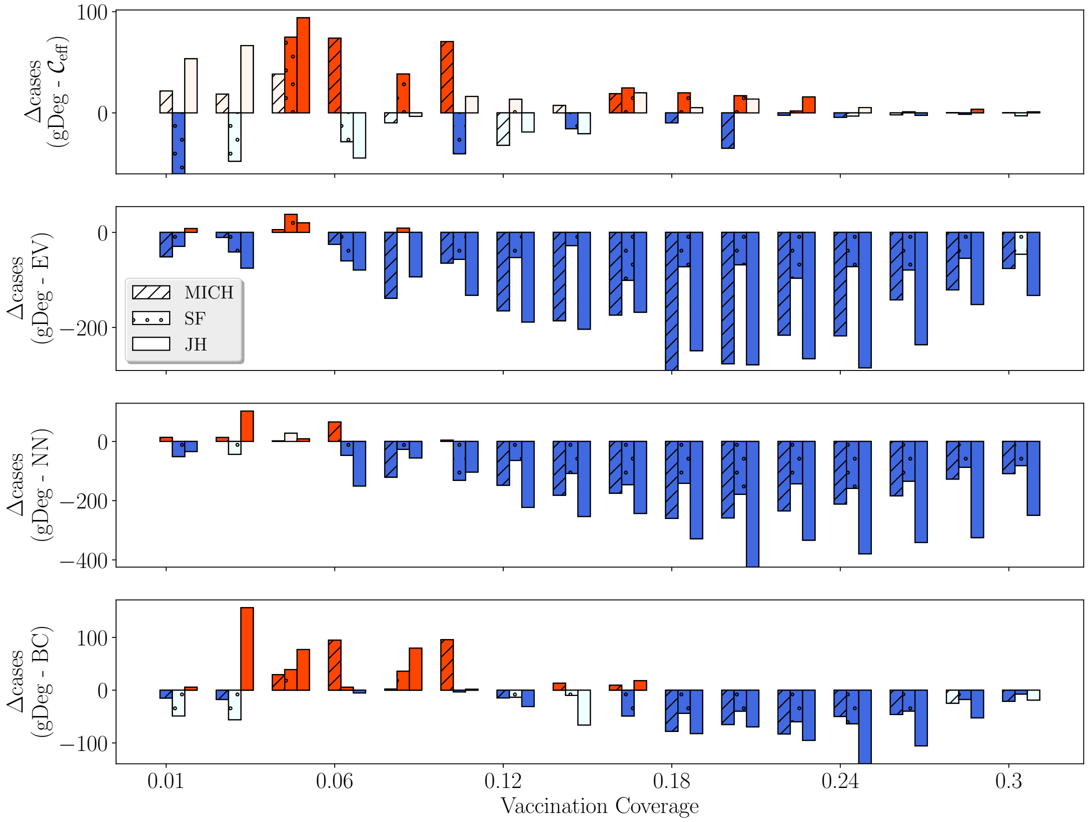


Figure 4.11: Comparing different immunization strategies with effective conductance, acquaintance, eigenvector centrality, and betweenness centrality with general degree ($r = 3$). The immunization coverage varies from %1 to %30 of the highest central nodes. Bars show the difference of final size of epidemic outbreak. Negative differences shows general degree performs better in the immunization compared to the other policy. By increasing the coverage, general degree outperforms other methods. Results are inferred by 2000 simulations of SIR epidemic model and statistically nonsignificant results are shown by shaded bars. Facebook friendship network of University of Michigan (MICH), UC San Francisco (SF) and Johns Hopkins (JH). Statistically insignificant differences are shown by shaded colors.

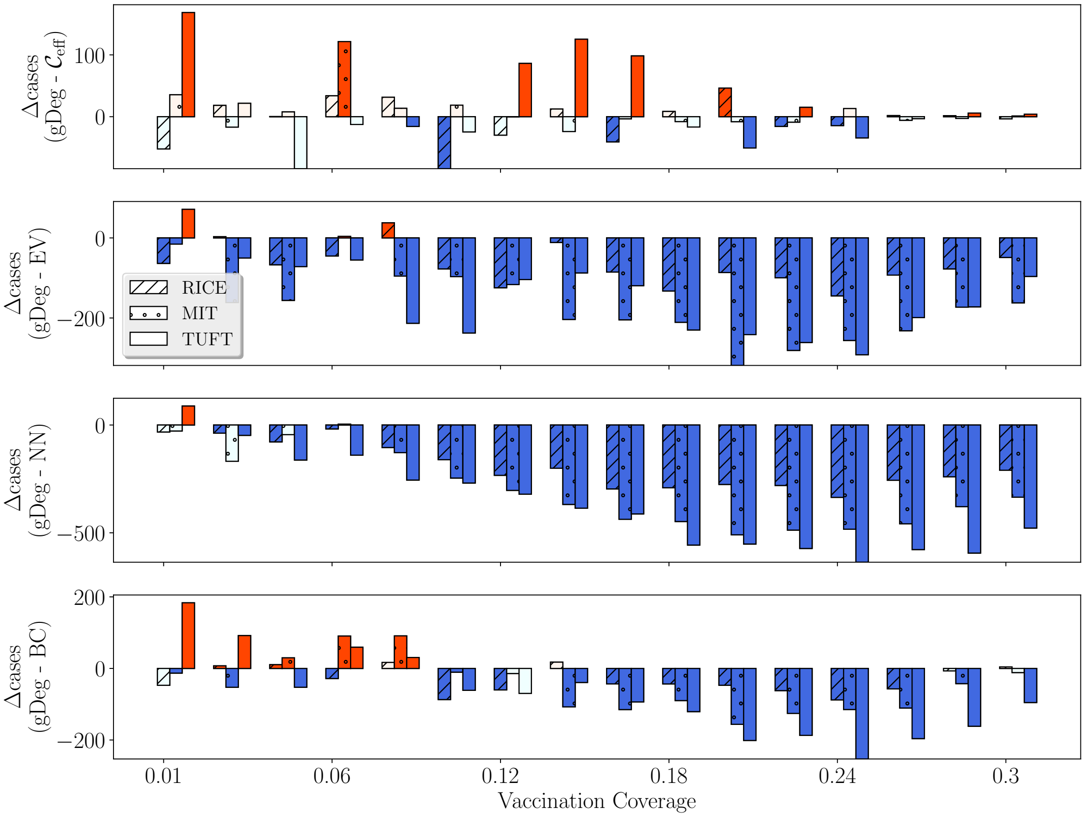


Figure 4.12: Comparing different immunization strategies with effective conductance, acquaintance, eigenvector centrality, and betweenness centrality with general degree ($r = 3$). The immunization coverage varies from %1 to %30 of the highest central nodes. Bars show the difference of final size of epidemic outbreak. Negative differences shows general degree performs better in the immunization compared to the other policy. By increasing the coverage, general degree outperforms other methods. Results are inferred by 2000 simulations of SIR epidemic model and statistically nonsignificant results are shown by shaded bars. Facebook friendship network of Rice University (RICE), Massachusetts Institute of Technology (MIT) and Tufts University (TUFT). Statistically insignificant differences are shown by shaded colors.

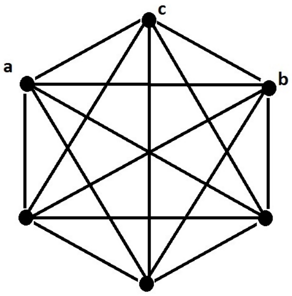


Figure 4.13: K_6- Complete graph on 6 nodes

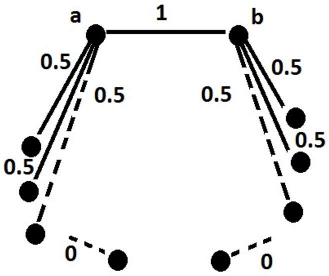


Figure 4.14: ρ^* for $\Gamma(a,b)$ on K_N

For convenience, for $j = 1, \ldots, n$, we write $A_j := V(K_j) \setminus \{x_{j-1}, y_j\}$, so that the shell at level j is $S_j = V(K_j) \setminus \{x_{j-1}\} = A_j \cup \{y_j\}$. Then, fix $m = 1, \ldots, n$, and for $j = 1, \ldots, m-1$, define the following density on $\in E(K_j)$:

$$\rho^*(e) := \begin{cases} \frac{1}{m} & \text{if } e = \{x_{j-1}, y_j\} \\ \\ \frac{1}{2m} & \text{if } e = \{x_{j-1}, a\} \text{ or } e = \{y_j, a\} \text{ for some } a \in A_j \\ \\ 0 & \text{otherwise} \end{cases}$$

For $j = m$, and $e \in E(K_m)$, set

$$\rho^*(e) := \begin{cases} \frac{1}{m} & \text{if } e = \{x_{m-1}, a\} \text{ for some } a \in A_m \cup \{y_m\} \\ \\ 0 & \text{otherwise} \end{cases}$$

Observe that the support of ρ^* can be decomposed as the disjoint union of $N-1$ paths. To see this, enumerate each $A_j = \{a_{j,k}\}_{k=1}^{N-2}$. Then, for $k = 1, \ldots, N-2$, let

$$\gamma_{m,k} := x_0 \ a_{1,k} \ x_1 \ a_{2,k} \ \cdots \ x_{m-1} \ a_{m,k}.$$

Finally set

$$\gamma_{m,0} := x_0 \ y_1 \ \cdots \ x_{m-1} \ y_m.$$

One can check that $\tilde{\Gamma} = \{\gamma_{m,k}\}_{k=0}^{N-2}$ is a Beurling subfamily for the shell modulus $\mathrm{Mod}_2(x_0, S_m)$. So

$$\mathrm{Mod}_2(x_0, S_m) = \frac{1}{m} + (N-2)\left[\frac{2m-2}{4m^2} + \frac{1}{m^2}\right] = \frac{N}{2m}\left(1 + \frac{1}{m}\right) - \frac{1}{m^2},$$

which is roughly $N/(2m)$. Also note that for $m = 1$ we recover the degree of x_0. If we sum we get

$$\sum_{m=1}^{n} \mathrm{Mod}_2(x_0, S_m) \simeq \frac{N}{2}\sum_{m=1}^{n}\frac{1}{m} \simeq \frac{N}{2}\log n.$$

The Ahlfors upper bound gives

$$\sum_{m=1}^{n} \frac{1}{\sum_{j=1}^{m} \frac{1}{N-1}} = (N-1)\sum_{m=1}^{n}\frac{1}{m} \simeq (N-1)\log n.$$

The generalized degree gives

$$\sum_{m=1}^{n} \frac{1}{m - 1 + \frac{1}{N-1}} \simeq N + \log n$$

Increasing sizes Now we repeat the construction above, but this time, setting $k_j := |V(K_j)|$, we have $k_1 = \alpha_1 + 2$ and, for $j = 2, \ldots, n$, we assume that $k_j = \alpha_j(k_{j-1} - 2) + 2$, for an increasing sequence of positive integers $\{\alpha_j\}_{j=2}^{n}$.

Then, fix $m = 1, \ldots, n$, and for $j = 1, \ldots, m-1$, define the following density on $\in E(K_j)$:

$$\rho^*(e) := \begin{cases} \frac{\prod_{k=j+1}^m \alpha_k}{1+\sum_{j=1}^m \prod_{k=j+1}^m \alpha_k} & \text{if } e = \{x_{j-1}, y_j\} \\ \\ \frac{2^{-1}\prod_{k=j+1}^m \alpha_k}{1+\sum_{j=1}^m \prod_{k=j+1}^m \alpha_k} & \text{if } e = \{x_{j-1}, a\} \text{ or } e = \{y_j, a\} \text{ for some } a \in A_j \\ \\ 0 & \text{otherwise} \end{cases}$$

For $j = m$, and $e \in E(K_m)$, set

$$\rho^*(e) := \begin{cases} \frac{1}{1+\sum_{j=1}^m \prod_{k=j+1}^m \alpha_k} & \text{if } e = \{x_{m-1}, a\} \text{ for some } a \in A_m \cup \{y_m\} \\ \\ 0 & \text{otherwise} \end{cases}$$

Now form $k_m - 1$ paths. Set

$$\gamma_{m,0} := x_0 \; y_1 \; \cdots \; x_{m-1} \; y_m.$$

As before, enumerate each $A_j = \{a_{j,k}\}_{k=1}^{k_j-2}$. Now, $k_m - 2 = \alpha_m(k_{m-1} - 2)$, so we can group the $k_m - 2$ edges $\{x_{m-1}, a\}$ for $a \in A_m$ into $k_{m-1} - 2$ groups of α_m edges. Each such group will then flow through a different node in A_{m-1}, and then we repeat. The claim is that this gives rise to a Beurling family of paths $\tilde{\Gamma}$. By construction, they all have ρ^* length equal to 1. We only need to check Beurling's criterion. So suppose $h \in \mathbb{R}^E$ satisfies

$$\ell_h(\gamma) \geq 0 \qquad \text{for all } \gamma \in \tilde{\Gamma}.$$

Then $\sum_{e\in E}\rho^*(e)h(e)$ is equal to:

$$\sum_{j=1}^{m}(\rho^*h)(x_{j-1},y_j)+\sum_{j=1}^{m-1}\sum_{i=1}^{k_j-2}[(\rho^*h)(x_{j-1},a_{j,k})+(\rho^*h)(a_{j,k},y_j)]+\sum_{i=1}^{k_m-2}(\rho^*h)(x_{m-1},a_{m,k}).$$

And if we write $\alpha := 1+\sum_{j=1}^{m}\prod_{k=j+1}^{m}\alpha_k$, and collect terms, this equals

$$\alpha^{-1}\left(\alpha\sum_{j=1}^{m}h(x_{j-1},y_j)+\sum_{j=1}^{m-1}\left(\prod_{k=j+1}^{m}\alpha_k\right)\sum_{i=1}^{k_j-2}[h(x_{j-1},a_{j,k})+h(a_{j,k},y_j)]+\sum_{i=1}^{k_m-2}h(x_{m-1},a_{m,k})\right).$$

which is ≥ 0, because for every $j=1.\ldots,m-1$

$$(k_j-2)\prod_{k=j+1}^{m}\alpha_k=k_m-2$$

So we get

$$\mathrm{Mod}_2(x_0,S_m)=\alpha^{-2}\left(1+\frac{3}{2}(k_m-2)\sum_{j=1}^{m}\prod_{k=j+1}^{m}\alpha_k+(k_m-2)\right)$$

Now choose $\alpha_j\equiv 2$. Then

$$\alpha=1+2+4+\cdots+2^{m-1}=2^m-1.$$

Also

$$k_m-2=2^{m-1}\alpha_1$$

So

$$\mathrm{Mod}_2(x_0,S_m)\simeq\alpha_1.$$

And

$$\sum_{m=1}^{n}\mathrm{Mod}_2(x_0,S_m)\simeq\alpha_1 n.$$

On the other hand the generalized degree is

$$\sum_{m=1}^{n} \frac{1}{m-1+\frac{1}{k_m-1}} \simeq \log n.$$

Chapter 5

Network clustering and community detection using modulus of families of loops

We study the structure of loops in networks using the notion of modulus of loop families. We introduce a new measure of network clustering by quantifying the richness of families of (simple) loops. Modulus tries to minimize the expected overlap among loops by spreading the expected link-usage optimally. We propose weighting networks using these expected link-usages to improve classical community detection algorithms. We show that the proposed method enhances the performance of certain algorithms, such as spectral partitioning and modularity maximization heuristics, on standard benchmarks.

This chapter is organized as follows. First, we introduce our notation and the necessary background on modulus of families of loops. We introduce an efficient algorithm to find the shortest weight cycle in graphs. Then, we define our proposed methods to measure clustering in the network. Next, we show how to preprocess a network in order to improve partitioning techniques such as Fiedler vector bisection and the modularity maximization heuristics. Finally, we discuss other potential applications.

5.1 Probability interpretation of loop modulus

We define a probability mass function $\mu \in \mathcal{P}(\mathcal{L}) := \{\mu \in \mathbb{R}^{\mathcal{L}}_{\geq 0} : \mu\mathbf{1} = 1\}$ that defines a random loop $\underline{\gamma} \in \mathcal{L}$ with

$$\mu(\gamma) = \Pr(\underline{\gamma} = \gamma). \tag{5.1}$$

Writing $\lambda = \nu\mu$ for a nonnegative scalar ν and a pmf μ (2.14) becomes:

$$\max_{\nu \geq 0} \left(\nu - \frac{\nu^2}{4} \min_{\mu \in \mathcal{P}(\mathcal{L})} \mu^T C \mu \right). \tag{5.2}$$

The maximum in (5.2) occurs when

$$\nu^* = 2 \left(\min_{\mu \in \mathcal{P}(\mathcal{L})} \mu^T C \mu \right)^{-1} \tag{5.3}$$

Substituting (5.3) in (5.2), we get that $\nu^* = 2\,\mathrm{Mod}_2(\mathcal{L})$ and

$$\mathrm{Mod}_2(\mathcal{L})^{-1} = \min_{\mu \in \mathcal{P}(\mathcal{L})} \mu^T C \mu = \mathbb{E}_{\mu^*} \left| \underline{\gamma_i} \cap \underline{\gamma_j} \right|,$$

for an optimal μ^*, where $\mathbb{E}_{\mu^*} \left| \underline{\gamma_i} \cap \underline{\gamma_j} \right|$ is the minimum expected overlap of two independent, identically distributed random loops with pmf $\mu^* \in \mathcal{P}(\mathcal{L})$.

Moreover by (2.13), the exremal density satisfies

$$\rho^*(e) = \mathrm{Mod}_2(\mathcal{L}) \mathbb{E}_{\mu^*} \left[\mathcal{N}(\underline{\gamma}, e) \right]$$

where $\mathbb{E}_{\mu^*} \left[\mathcal{N}(\underline{\gamma}, e) \right] = \sum_{\gamma \in \mathcal{L}} \mathcal{N}(\gamma, e) \mu^*(\gamma)$ is the expected usage of link e in loop $\underline{\gamma}$. Therefore, the optimal measures μ^* are related to the optimal density ρ^* as follows:

$$\frac{\rho^*(e)}{\mathrm{Mod}_2(\mathcal{L})} = \mathbb{P}_{\mu^*} \left(e \in \underline{\gamma} \right) \tag{5.4}$$

We call $\mathbb{P}_{\mu^*} \left(e \in \underline{\gamma} \right)$ the *expected usage* of link e.

Moreover, one can always find an optimal measure μ^* that is supported on a minimal set

Algorithm 3 Approximating densities for $\text{Mod}_2(\mathcal{L})$ with tolerance $0 < \epsilon_{\text{tol}} < 1$[50]

```
$\rho \leftarrow 0$; $\rho_0 \leftarrow \mathbf{1}$
$\mathcal{L}' \leftarrow \emptyset$
$\gamma \leftarrow \mathit{ShortestLoop}(\rho_0)$
**while** $\exists \gamma$ such that $\ell_\rho(\gamma) \leq 1 - \epsilon_{\text{tol}}$ **do**
    $\mathcal{L}' \leftarrow \mathcal{L}' \cup \{\gamma\}$
    $\rho \leftarrow \text{argmin}\{\mathcal{E}_2(\rho) : \mathcal{N}\rho \geq \mathbf{1}\}$
**end while**
```

of loops of cardinality bounded above by $|E|$, see[52] Theorem 3.5. We think of these loops as "important loops" that play a role in the optimization problems as active constraints.

5.2 Approximating the modulus

The numerical results in the examples that follow are produced by a Python implementation of the simple algorithm described in[50]. This algorithm exploits the $\mathcal{L}$-monotonicity (Property (b)) of the modulus by building a subset $\mathcal{L}' \subseteq \mathcal{L}$ so that $\text{Mod}_2(\mathcal{L}') \approx \text{Mod}_2(\mathcal{L})$ to a desired accuracy[50] Theorem 9.1. In short, the algorithm begins with $\mathcal{L}' = \emptyset$, for which the choice $\rho \equiv 0$ is optimal and insert a loop with the shortest hop-length then repeatedly adds violated constraints to $\mathcal{L}'$ and determines the optimal ρ each time. The algorithm terminates when all constraints are satisfied to a given tolerance (Algorithm 1).

The two key ingredients for implementing this algorithm are a solver for the convex optimization problem (4.2) and a method for finding violated loops, i.e., with ρ-length less than one. In our implementation, the optimization problem is solved using an active set quadratic program[63] and the violated constraint search is performed using a modified version of the breadth-first search from each node that has a cut-off $1-$tol and reports the first backward link that forms a loop less than the cut-off.

Although simple, this algorithm is adequate for computing the modulus in the examples presented here, on a Linux operating computer with Intel core i7 (and 2.80 GHz base frequency) processor, for example. More advanced parallel primal-dual algorithms are currently under development to treat modulus computations on larger networks.

5.3 Finding the shortest loop

Finding shortest cycle in graphs is a fundamental problem, but less explored compared to finding the shortest path. Efficient algorithm to find shortest cycle of a graph, also known as girth[89], is critical in cycle theory, determination of minimal cycle basis[90–92] and maximum cycle packing[93–95]. In particular, efficiently finding girth is crucial for methods that iteratively populate all loops with selected weights[?] . The shortest cycle problem is also related to graph properties such as chromatic number and connectivity[96;97], also for planar graphs it corresponds to the min-cut problem in the dual graph.

We can use all pair shortest path algorithm (Floyd-Warshall) to find the shortest cycle in directed graphs. However, this procedure is not directly applicable on undirected graphs due to possible self-loops and the objective of finding simple cycles without repetition of edges. For unweighted graphs Itai *et al.* used reductions in subcubic time (upperbounded by the matrix multiplication exponent); they left the weighted graphs[98] as an open problem. Roditty *et al.* extended Itai *et al.* result for integer weights[99].

Vassilevska *et al.* relates finding minimum cycle to other graph theory problems with no subcubic time algorithm[100] and thus any improvements for one of these problems influences others.

Further approximation algorithms are proposed in[98;101–104]. Finding shortest cycle with even length is analyzed in[105], and randomized algorithms proposed, e.g., in[103]. In this section, we focus on a deterministic algorithm to find the girth that has arbitrary length.

Because more efficient combinatorial algorithm is required for (real valued) weighted and undirected graphs, this research introduces an easily implementable method that incorporates the known shortest path algorithm philosophy. We employ a unique definition for walks with sockets (pair consisting of a vertex and an edge) and modify the existing Dijkstra's algorithm respectively. We translate the algorithm into nodes and edges afterwards.

Most of the proposed algorithms are focused on finding the shortest cycle rooted to each node and repeating the process for all nodes, e.g. in[98;103;106]. Nevertheless, we focused on minimizing a *composite distance* from each node to the cycles, with shrinking the network

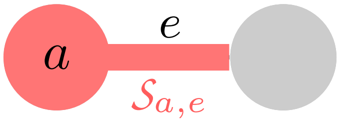


Figure 5.1: Socket $\mathcal{S}_{a,e}$ consisting of vertex a and edge e.

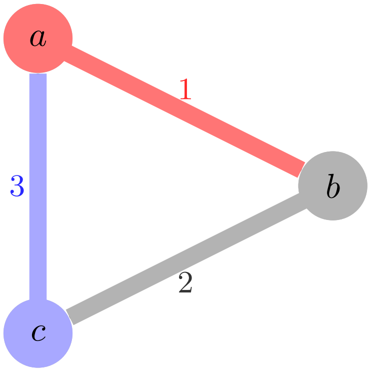


Figure 5.2: Chain of sockets $\mathcal{S}_{a,1}\mathcal{S}_{b,2}\mathcal{S}_{c,3}$ form a loop on vertices a, b, and c.

with discarding nodes that are not improving the subsequent searches.

Let $G = (V, E)$ be a graph with vertex set V and edge set E. The size of V and E are denoted by n and m subsequently. The set of neighbors of node v is denoted by $N_v = \{u \in V | u \text{ is connected to } v\}$. A *walk* γ on a graph is represented by a finite string of vertices and edges $v_1 e_1 v_2 e_2 \ldots v_r$, where v_i's and e_i's are all distinct. A *cycle* is a walk that starts and ends at the same vertex $v_1 e_1 v_2 e_2 \ldots v_r e_r v_1$. The length of cycle c is defined as

$$\ell(c) := \sum_{e \in c} w(e), \tag{5.5}$$

where $w : E \to \mathbb{R}^+$ is a weight function, interpreted as length of edges.

A *socket* $\mathcal{S}_{v_i,e_j}$ includes a connected pair of vertex and edge (v_i, e_j), as shown in Figure 5.1. *Distinct sockets*, have no nodes or edges in common and weight of a socket $\mathcal{S}_{v_i,e_j}$ is weight of e_j.

Therefore, walk can be redefined with sockets; a finite string of distinct sockets that do not share vertices and edges, i.e., $\mathcal{S}_{v_1,e_1}\mathcal{S}_{v_2,e_2} \ldots \mathcal{S}_{v_r,e_r}$ is a simple walk, and a cycle is a simple walk that starts from an initial socket $\mathcal{S}_{v_1,e_1}$ and ends in $\mathcal{S}_{v_r,e_r}$, where e_1 and e_r are two distinct edges incident to vertex v_1. A simple loop with sockets is shown in Figure 5.2.

5.3.1 Composite distance

Let $\mathcal{L}$ be the set of simple cycles in G; assume $\mathcal{L}$ is nonempty, i.e., G is not a tree. For $x, y \in V$ and $c \in \mathcal{L}$, let $d(x, y)$ be the distance between x and y, and $d(x, c)$ be the distance between x and c, that is

$$d(x, c) = \min_{y \in c} d(x, y).$$

Additionally, define the composite distance of node $x \in V$ to loop $c \in \mathcal{L}$:

$$d^+(x, c) = d(x, c) + \ell(c), \tag{5.6}$$

and the composite distance of node x to all loops $\mathcal{L}$:

$$d^+(x) = \min_{c \in \mathcal{L}} d^+(x, c). \tag{5.7}$$

The following theorem shows that we obtain the shortest cycle by solving the optimization problem (5.7) for every vertex x,

Theorem 5.3.1. *Minimizing $d^+(x)$ over $x \in V$ is equivalent to finding the shortest cycle in $\mathcal{L}$, and the minimum is attained for any x in a shortest cycle.*

Proof. For any $x \in V$ and $c \in \mathcal{L}$, $d^+(x, c) \geq \ell(c)$, and equality holds if $x \in c$, so the minimum will be attained when c is a shortest cycle and $x \in c$. □

Theorem 5.3.1 suggests to find a shortest cycle, we determine $d^+(x)$ for all nodes subsequently, with using the previous best $d^+(x)$ as a cut-off for the next. The following theorem shows that we can exclude node z that $d(x, z) \leq d(x, c)$ from further consideration in our search.

Theorem 5.3.2. *Suppose $d^+(x) = d^+(x, c)$ for some $x \in V$ and $c \in \mathcal{L}$. Let $c' \neq c \in \mathcal{L}$ be a shortest cycle and let $y \in c'$. Then*

$$d(x, y) > d(x, c).$$

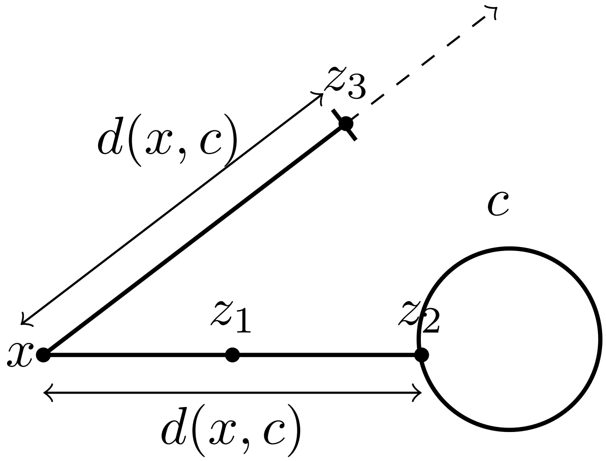


Figure 5.3: Nearest cycle to node x. Nodes such as z_1, z_2, and z_3 that $d(x, z_i) \leq d(x, c)$ can be excluded from further considerations.

In other words, after a search starting from x, we can exclude from further consideration any node that x is closer to them than the minimizing cycle c.

Proof. Since $d^+(x) < d^+(x, c')$ and $\ell(c') < \ell(c)$, we have

$$d(x, c) + \ell(c) < d(x, c') + \ell(c') < d(x, c') + \ell(c),$$

so

$$d(x, c) < d(x, c') < d(x, y),$$

implying the theorem. □

As a trivial example for Theorem 5.3.2, after a search from node v, all nodes z that $d(x, z) \leq d(x, c)$ will be excluded from subsequent searches (Figure 5.3).

5.3.2 Algorithm

The proposed method searches the sockets in the graph with Dijkstra's algorithm while using a priority queue implementation to map each socket to its position in the queue[107]. We illustrate an example of a cycle including vertex a with degree 3 that transforms into a cycle in Figure 5.4(a) to a path of distinct sockets from set $\{\mathcal{S}_{a,1}, \mathcal{S}_{a,2}, \mathcal{S}_{a,3}\}$ to $\{\mathcal{S}_{N_1,1}, \mathcal{S}_{N_2,2}, \mathcal{S}_{N_3,3}\}$

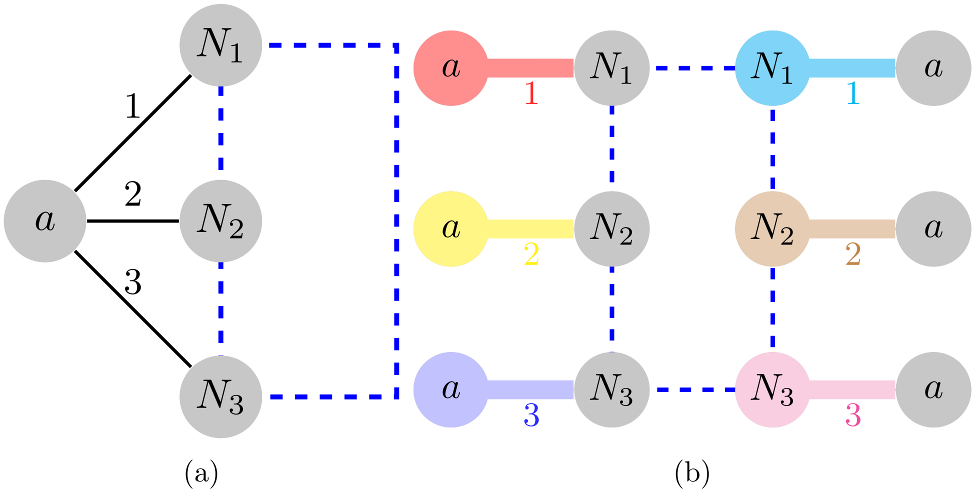


Figure 5.4: (a) Vertex a with neighbors. The remainder of the graph is shows by a dashed rectangle; (b) shortest path between set of sockets $\{\mathcal{S}_{a,1}, \mathcal{S}_{a,2}, \mathcal{S}_{a,3}\}$ to another set of sockets $\{\mathcal{S}_{N_1,1}, \mathcal{S}_{N_2,2}, \mathcal{S}_{N_3,3}\}$, such that all sockets are distinct, gives the shortest cycle for vertex a.

in Figure 5.4(b).

Following Theorem 5.3.1, to find a shortest cycle we examine the smallest found $d^+(x)$ over $x \in V$ by running Dijkstra's algorithm (see Algorithm 4). Whence the shortest path starting from sockets attached to x meets a nondistinct socket, algorithm returns $d^+(x)$.

We illustrate an example of a weighted graph in Figure 5.5(a) and the found $d^+(a)$ in Figure 5.5(b) with nodes that can be excluded from further consideration in yellow color. We demonstrate different steps of the algorithm in Figure 5.6.

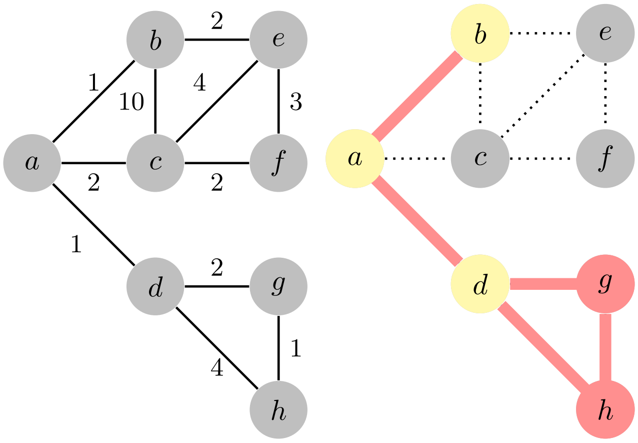
To find the girth we apply Algorithm 4 for each node as root and use the shortest found cycle as a cut-off for the next search with excluding the nodes that cannot be roots for future searches using Theorem 5.3.2. The pseudo-code is shown in Algorithm 5, from the algorithms and theorems we conclude the following corollary:

Corollary 5.3.3. *Using Algorithm 4 and 5, each loop will be searched at most only once. Moreover, the number of loops that the algorithm completes is upperbounded by size of cycle basis*[108].

We translate the proposed algorithms into vertices and edges. Following Theorem 5.3.1, to find the girth we examine the smallest found $d^+(x)$ over $x \in V$ exhaustively using Dijkstra's algorithm: when a node z is added to the set of "visited" nodes (i.e., the nodes whose a

Algorithm 4 Algorithm to determine improved $d^+(x) = \min_{c\in C} d^+(x,c)$ and excluded sockets.

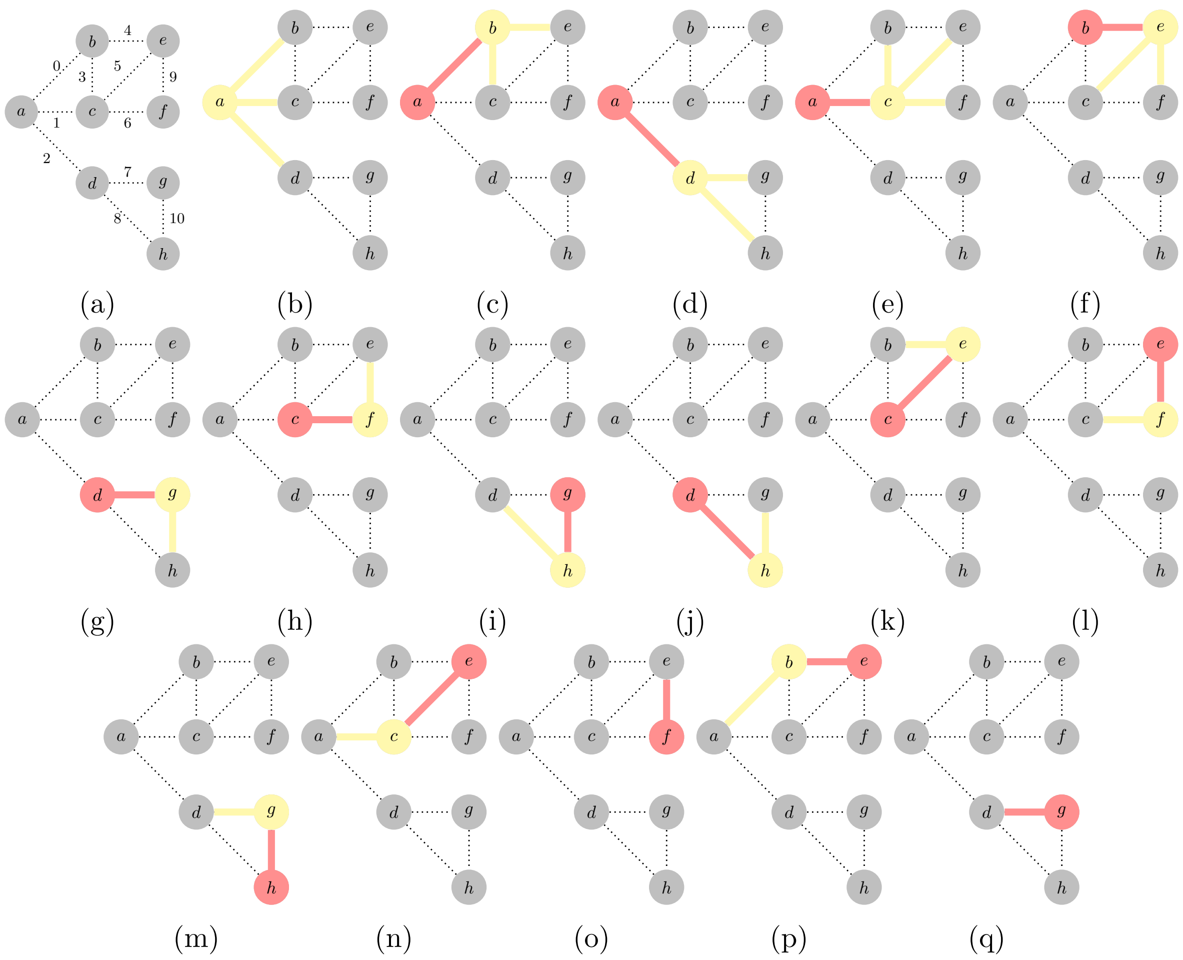
```
$Q$.**enqueue**(Adjacent sockets to $x$ with their weights)
walk$\leftarrow$ Dictionary of shortest walk for each socket from source sockets
**while** $Q$ **do**
    $\mathcal{S}$, dist($\mathcal{S}$) $\leftarrow$ $Q$.**dequeue**
    **if** dist($\mathcal{S}$) $>$ Cut-off **then**
        **return** None
    **end if**
    $u \leftarrow$ adjacent vertex to $\mathcal{S}$.
    **if** $u \in$ socket $\mathcal{T}$ in walk[$\mathcal{S}$] **then**
        **return** dist($\mathcal{S}$) and nodes in sockets with distance less than dist($\mathcal{T}$).
    **end if**
    **for** Sockets $\mathcal{R}_{u,e_i}$ starting from $u$ and distinct from $\mathcal{S}$ **do**
        dist($\mathcal{R}_{u,e_i}$) = dist($\mathcal{S}$) + weight of $w(e_i)$)
        **if** dist($\mathcal{R}_{u,e_i}$) $>$ Cut-off **then**
            continue to next iteration
        **end if**
        **if** $\mathcal{R}_{u,e_i} \notin$ seen or dist($\mathcal{R}_{u,e_i}$) <seen[$\mathcal{R}_{u,e_i}$] **then**
            seen[$\mathcal{R}_{u,e_i}$] $\leftarrow$ dist($\mathcal{R}_{u,e_i}$)
            update walk[$\mathcal{R}_{u,e_i}$] = walk[$\mathcal{S}$] + $\mathcal{R}_{u,e_i}$
            $Q$.**enqueue**($\mathcal{R}_{u,e_i}$, dist($\mathcal{R}_{u,e_i}$))
        **end if**
    **end for**
**end while**
**return** None
```



(a)

(b)

Figure 5.5: (a) Example graph with weights. (b) Nearest loop to vertex a, i.e., $\arg\min_{c\in C} d^+(x,c)$. Sockets linked to nodes a, b, and d will be discarded for the subsequent searches.



(a) (b) (c) (d) (e) (f)

(g) (h) (i) (j) (k) (l)

(m) (n) (o) (p) (q)

Figure 5.6: Solution steps in which each yellow socket is pushing into the queue and red sockets are popping. (a) Enumerating edges for better explanation (unnecessary for the proposed algorithm); (b) pushing the immediate attaching socket to vertex a with their weights as priority: $(\mathcal{S}_{a,0}, 1)$, $(\mathcal{S}_{a,2}, 1)$, and $(\mathcal{S}_{a,1}, 2)$; (c) $(\mathcal{S}_{a,0}, 1)$ popped and neighboring sockets with new distances pushed: $(\mathcal{S}_{b,3}, 11)$ and $(\mathcal{S}_{b,4}, 3)$; (d) $(\mathcal{S}_{a,2}, 1)$ popped, and $(\mathcal{S}_{d,7}, 3)$, and $(\mathcal{S}_{d,8}, 5)$ pushed; (e) $(\mathcal{S}_{a,1}, 2)$ popped and $(\mathcal{S}_{c,3}, 12)$, $(\mathcal{S}_{c,5}, 6)$, and $(c_{c,6}, 4)$ pushed. (f) $(\mathcal{S}_{b,4}, 3)$ popped and $(\mathcal{S}_{e,5}, 7)$, and $(\mathcal{S}_{e,9}, 6)$ pushed; (g) $(\mathcal{S}_{d,7}, 3)$ popped and $(\mathcal{S}_{g,10}, 4)$ pushed; (h) $(c_{c,6}, 4)$ popped and $(\mathcal{S}_{f,9}, 7)$ pushed; (i) $(\mathcal{S}_{g,10}, 4)$ popped and $(\mathcal{S}_{h,8}, 8)$ pushed; (j) $(\mathcal{S}_{d,8}, 5)$ popped and $(\mathcal{S}_{h,10}, 6)$ pushed; (k) $(\mathcal{S}_{c,5}, 6)$ popped and $(\mathcal{S}_{e,4}, 8)$ pushed; (l) $(\mathcal{S}_{e,9}, 6)$ popped and $(\mathcal{S}_{f,6}, 8)$ pushed; (m) $(\mathcal{S}_{h,10}, 6)$ popped and $(\mathcal{S}_{g,7}, 8)$ pushed; (n) $(\mathcal{S}_{e,5}, 7)$ popped and $(\mathcal{S}_{c,1}, 9)$ pushed; (o) $(\mathcal{S}_{f,9}, 7)$ popped but does not lead to any socket from a previously shorter walk; (p) $(\mathcal{S}_{e,4}, 8)$ popped and $(\mathcal{S}_{b,0}, 9)$ pushed; (q) $(\mathcal{S}_{f,6}, 8)$ popped and interrupts its shortest path. The algorithm returns the closest circle: $\mathcal{S}_{d,8}, \mathcal{S}_{h,10}, \mathcal{S}_{g,7}$ with length $4 + 1 + 2 = 7$, shown in Figure 5.5(b).

Algorithm 5 Using Algorithm 4 to find the girth

```
$cut\text{-}off \leftarrow \infty$
while $V$ do
    $v \leftarrow V.pop$
    $c$, $Y \leftarrow$ Algorithm 4 for $v$, and with $cut\text{-}off$
    remove $Y$ from $G$
    if $cut\text{-}off > \ell(c)$ then
        $cut\text{-}off = \ell(c)$
    end if
end while
return $cut\text{-}off$
```

path from x to them is now determined). For each neighbor y of z, if a shorter path is found from the root, we update the shortest walk to y. If y has been visited and the found shortest path to y cannot be improved, we have a candidate for $d^+(x)$ of the root node x, namely the walk that begins at x, follows the Dijkstra tree to z crosses $\{z, y\}$ and returns to x along the Dijkstra tree. The candidate composite distance is $d(x, z) + d(x, y) - d(x, ca) + w(\{x, y\})$, where ca is the common ancestor of z and y in the shortest path tree. Algorithm enqueues the found candidate for the composite distance to the priority queue. Whenever a candidate dequeues by the algorithm, a composite distance $d^+(x)$ is found, finishing the search. As we run Dijkstra, we keep track of the best $d^+(x)$ found so far as a cut-off for the next search. If, at any point, we visit a node z that is farther from x than this best distance, we stop the search and excludes nodes that are not improving further searches by Theorem 5.3.2.

In the following theorem, we obtain the algorithm complexity.

Theorem 5.3.4 (Complexity of the algorithm)**.** *The worst case complexity for finding the girth from*

$$\min_{x\in V}\left(\min_{c\in C} d^+(x, c)\right) \tag{5.8}$$

with algorithms 4 and 5 using socket is $\mathcal{O}(\langle k\rangle n^2 \log n)$, where $\langle k\rangle$ is the average degree of the network and with nodes and edges is in $\mathcal{O}(nm + n^2 \log n)$ or.

Proof. In the socket language; from a node v, algorithm starts searching from all adjacent sockets and at most completes a tree with an extra socket to close a loop, i.e., $n - 1$ sockets plus one. In each step i, from Theorem 5.3.2, algorithm excludes nodes that cannot improve the found shortest cycle and thus the network is shrinking. Therefore, the total number of heap operations is

$$\sum_{i=1}^{n} k^*(i)(n-i+1) \tag{5.9}$$

where $k^*(i) : \{1 : N\} \to \mathbb{Z}$ is the degree at step i. Equation (5.9) is bounded by $nk_{\max}$ because

$$\sum_{i=1}^{n} k^*(i)(n-i+1) < n\sum_{i=1}^{n} k^*(i) = nm$$

Therefore the worst case complexity, with enqueuing heap operation in $\mathcal{O}(\log(n))$, is in $\mathcal{O}(\langle k \rangle n^2 \log n)$. With nodes and edges it is in $\mathcal{O}(nm + n^2 \log n)$ □

5.3.3 Examples

We consider test examples to compare the algorithm with the naive counterparts where to find the girth in the graph is for all edges $e = \{u, v\} \in E$, we find the shortest path γ from u and v, such that $e \notin \gamma$, the resultant cycle $\gamma + e$, is the shortest cycle rooted to e. Repeating this process for all edges and comparing the length of them results in choosing the shortest one in $\mathcal{O}(m^2 + nm \log n)$ using Dijkstra's algorithm with Fibonacci heap operations.

In the first example, we consider a random geometric graph to illustrate the algorithm behavior. Moreover, because complete graphs (or similar graphs) are posing a challenge for the algorithm performance due to their large degrees, we test them as the second example.

Random geometric graph with light spanning tree An example for short-circuiting in networks can be described in a random geometric network G, with weight values chosen uniformly random between 10^4 and 10^5. After finding a spanning tree $T = (V, E_T)$, we reweigh the edges E_T to be uniformly random between 20 and 50. Now if we randomly choose one edge in $E \setminus E_T$ and assign a small weight, the shortest cycle comprises this edge and some edges in E_T. Finding this cycle is hard for the naive algorithm because it finds the shortest cycle rooted to each edge. However, our proposed algorithm finds this shortest

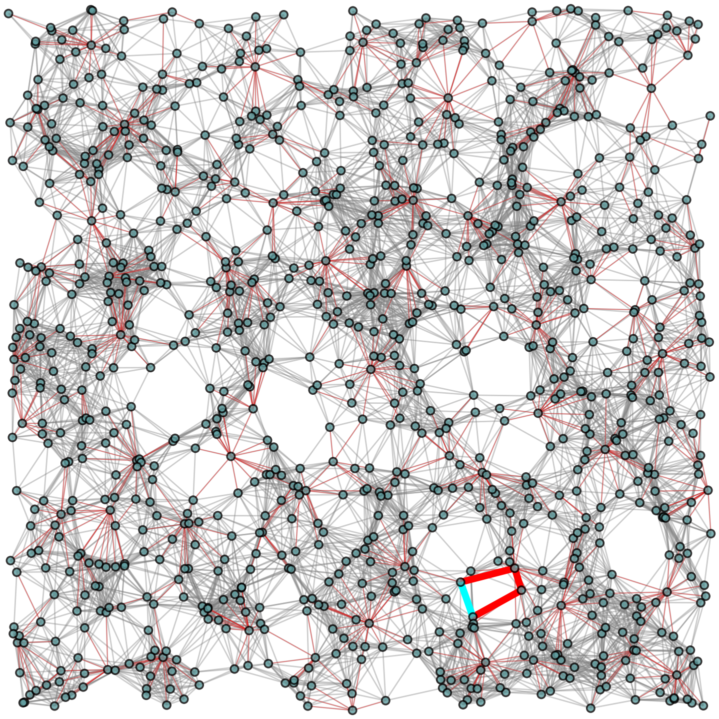


Figure 5.7: Shortest cycle found in a spatial network, with small weights on the spanning tree and one nontree edge.

cycle quickly (here with one iteration), see Figure 5.7 for the illustration.

Complete network We plot the total number of heap operations in the naive algorithm and our proposed algorithm for several randomly weighted complete graph instances in Figure 5.8. The proposed algorithm outperforms the naive counterpart with shrinking the network in each search.

In summary, we proposed a deterministic algorithm to find shortest cycle in graphs. Instead of finding the shortest cycle rooted at each node, we focus on finding the shortest composite distance of a node to cycles in the graph. We proved that algorithm is in $\mathcal{O}(nm)$. Another way to right the worst case complexity is with the average degree $\langle k \rangle$ in

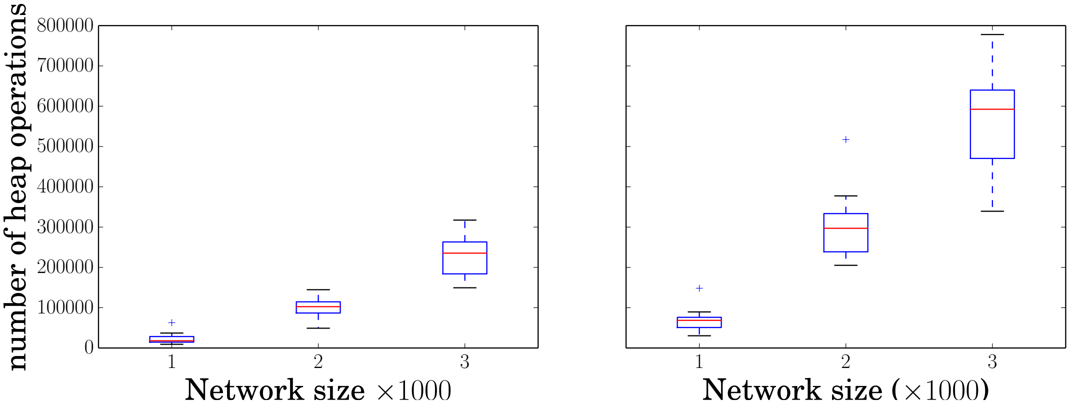


Figure 5.8: Heap operations in our algorithm (left) compared to the naive algorithm (right) for complete graphs with random positive weights.

$\mathcal{O}(\langle k \rangle n^2 \log n)$ and thus subcubic when graphs average degree is in $\mathcal{O}(n^{1-\epsilon}$ which often is the case in empirical applications.

5.4 Clustering measure with modulus of family of loops

Complex networks exhibit properties such as the small-world phenomenon[11], scale-free degree distribution[6], and local clustering of nodes[11]. In social networks, when two individuals are acquainted it is probable that they have another friend in common, resulting in properties of homophily for the network. For example, in friendship networks people introduce their friends to each other. This transitivity property makes the real world networks different from synthetic random networks[109]. However, this clustering tendency is difficult to quantify.

A proposed measure of clustering for a node v[11] is to compute the fraction of links between neighbors of v that actually are in the network, over all possible ones. The authors in[110] pointed out the importance of closed paths (loops) in the cluster and discussed computation of the clustering coefficient using the density of loops with length 3 (triangles). Because this measure fails to describe the clustering of grid-like parts of the network, the authors improved the measure by counting quadrilaterals–loops with length 4 or *mutuality* in[109]– and proposed a new measure that considers different types of quadrilaterals. Similarly,[47] addresses bipartite networks that lack triangles thus the standard clustering coefficient is not useful. In[47],[111] and[112] the authors emphasize the importance of longer loops in the network. The authors in[113] showed that clustering coefficient measures are highly correlated with degree, and they proposed a measure that preserves the degree sequence for the maximum possible links among neighbors of node v, thus avoiding correlation biases. Kim *et al.* introduced *local cycling coefficient* that quantifies local circle topologies by averaging the inverse length of loops passing the nodes[49]. They average this coefficient for all nodes to derive the degree of circulation in the network.

The authors in[114] introduced a version of clustering coefficient that considers weighted network, and[115] propose a way to measure a general clustering coefficient for weighted and directed networks.

Numerous versions of clustering coefficients for different types of networks expose the need for a generalized measure that works for a wide range of applications. We apply the concept of modulus of families of loops as a tool to study structural properties of network clustering. In this section, we show that analysis of loops using modulus provides a general approach to the study of network clustering properties. We also propose a new clustering measure that can explain situations that conventional methods struggle to handle.

A network has a high clustering measure when most of the links are included in short loops that also visit nearby links. The standard method of counting triangles considers the smallest loops, while other methods consider the next shortest loops, i.e., quadrilaterals. A method must be devised to compare these loops and evaluate the combined influence to improve clustering measures[109]. The previous section introduced a way to evaluate family of loops using modulus. Therefore, we propose a comprehensive modulus-based measure of clustering.

The classical clustering coefficients that measure triangle density, are usually normalized by comparing the links in the networks (that form triangles) with all possible links between nodes, i.e., all possible triangles in the corresponding complete graph. Most real networks are far from being complete graphs (even locally), therefore, classical coefficients usually have small values, and they are correlated to the degree of the node[113].

We normalize our clustering measure using the probabilistic interpretation in (5.4). Modulus tries to spread expected usage as much as possible among the links of the network in order to minimize the expected overlap. However, the expected link usages are not always uniform. Define a uniform density $\rho_u(e) \equiv 1/3$ that is always admissible for loop modulus–because it penalizes all loops at least 1. So its energy $\mathcal{E}_2(\rho_u) = |E|/9$ gives an upper bound for $\mathrm{Mod}_2(\mathcal{L})$.

Therefore, our proposed clustering measure takes the following form

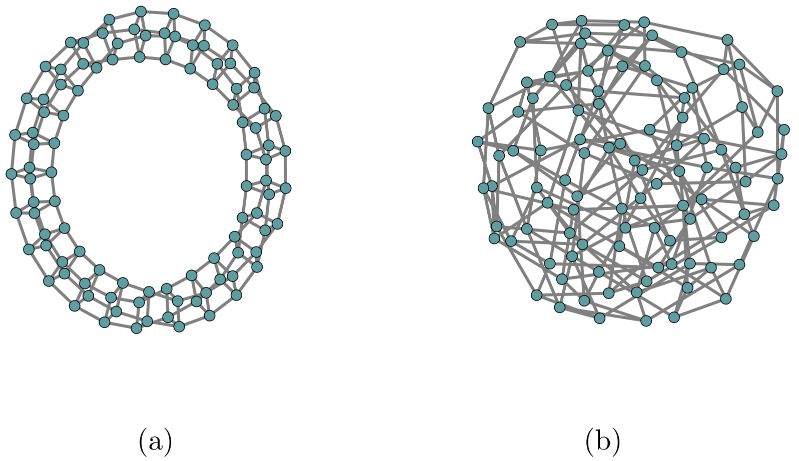
$$C_{\mathrm{loop}}(G) := \frac{9}{|E|}\,\mathrm{Mod}_2(\mathcal{L}), \tag{5.10}$$

where C_{loop} is a measure of richness of actual link participation in important loops over the ideal case that all links participate equally in triangles. For example, consider a grid as in Figure 5.9(a) with 100 nodes and 200 links. We compare its loop modulus with that of a random regular network with the same number of nodes and same degree as shown in Figure 5.9–these networks behave similar to the two extremes of small world networks[11]. Since the classical methods use the number of triangles in a network, they give zero clustering coefficient to the grid and $2-3\%$ to the random regular network. The grid has square clustering coefficient 14.7% and the random regular network square clustering is close to zero (we use square clustering introduced in[47]). For each network in Figures 5.9(a) and 5.9(b):

$$\mathrm{Mod}_2\,\mathcal{L}_{\mathrm{grid}} = 10.8 \quad \text{and} \quad \mathrm{Mod}_2\,\mathcal{L}_{\mathrm{reg}} = 7.8.$$

Therefore, $C_{\mathrm{loop}}\left(G_{\mathrm{grid}}\right) = 54\%$ which means the network is highly clustered and $C_{\mathrm{loop}}\left(G_{\mathrm{reg}}\right) = 34\%$ is less clustered than grid.

In some cases, our proposed measure gives different conclusions than the classical cluster coefficients. For example, let us compare the networks (a) and (b) in Figure 5.10. Network (a) is collaboration network between Jazz musicians[14] and network (b) is an email communication network at the University Rovira i Virgili in Spain[15]. In the email communication network a very rich core is balanced by many stems on the periphery and the loop clustering measure is slightly higher than for the Jazz network. This goes in the opposite direction



(a) (b)

Figure 5.9: (a) A grid network with deg = 4 and 100 nodes, (b) a random regular network with deg = 4 and 100 nodes. The proposed clustering measure is $C\left(G_{\text{grid}}\right) = 56.25\%$, $C\left(G_{\text{reg}}\right) = 34\%$. Classical clustering coefficient gives zero for the grid and 2.4% for the regular network and average square clustering coefficient is 14.7% for the grid and 0.4% for the regular network.

.

than the classical clustering coefficient result[116]. For the piece of the Facebook network in Figure 5.10(c)[3], the loop clustering value is slightly greater than the classical case, reflecting a certain amount of tightly knit communities. Finally, in the friendship network for the website hamsterster[16], the clustering measure and classical clustering coefficient give almost similar results.

Furthermore, we can isolate the contribution of triangles, squares, and higher order loops by considering modulus of subfamilies of $\mathcal{L}$. This can be done assuming a hop-length cut-off for γ in Algorithm 4. Moreover, the property of subadditivity (Property (e)) gives an upperbound for the aggregate effects.

5.5 Weighting to enhance community detection algorithms

Communities in networks are defined as groups of nodes that are closely knit together relative to the rest of the network. Real world networks, for example social networks[117] and biological

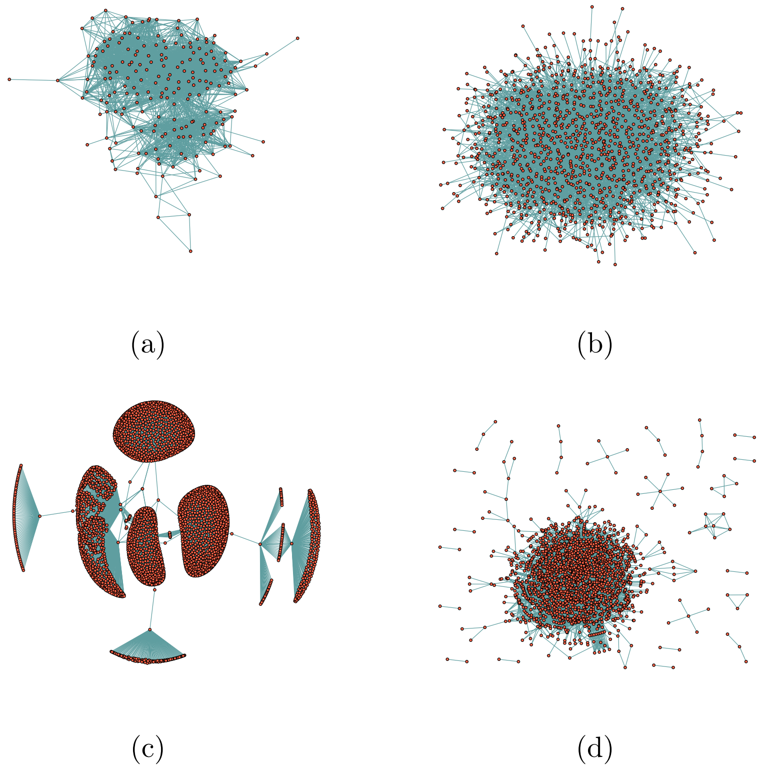

(a)
(b)
(c)
(d)

Figure 5.10: (a) Jazz musicians network[14] with $C_{\text{loop}} = 10.0\%$; average triangle density $C = 52.0\%$ and average square clustering 6.66%. (b) Email communication network in University Rovira i Virgili in Spain with $C_{\text{loop}} = 13.8\%$; average triangle density $C = 16.6\%$ and average square clustering 1.46%[15]. (c) An excerpt of Facebook network with $n = 2888$ and $m = 2981$. Edges represent friendships between nodes[3] with $C_{\text{loop}} = 3.7\%$; average triangle density 0.03% and average square clustering 0.07%. (d) Friendship network of the website hamsterster.com[16], with $n = 1858$ and $m = 12534$. The clustering in the network is $C_{\text{loop}} = 6.22\%$. The classical clustering coefficient (transitivity) is 9.04% and average square clustering coefficient 6.78%.

networks[118], comprise densely connected parts that are loosely connected with each other. Finding these communities is crucial in analyzing the collective behavior of the network or in order to be able to make assumptions (meta population). These communities can be disjoint or overlapping. For a comprehensive review of the literature on this subject see[28].

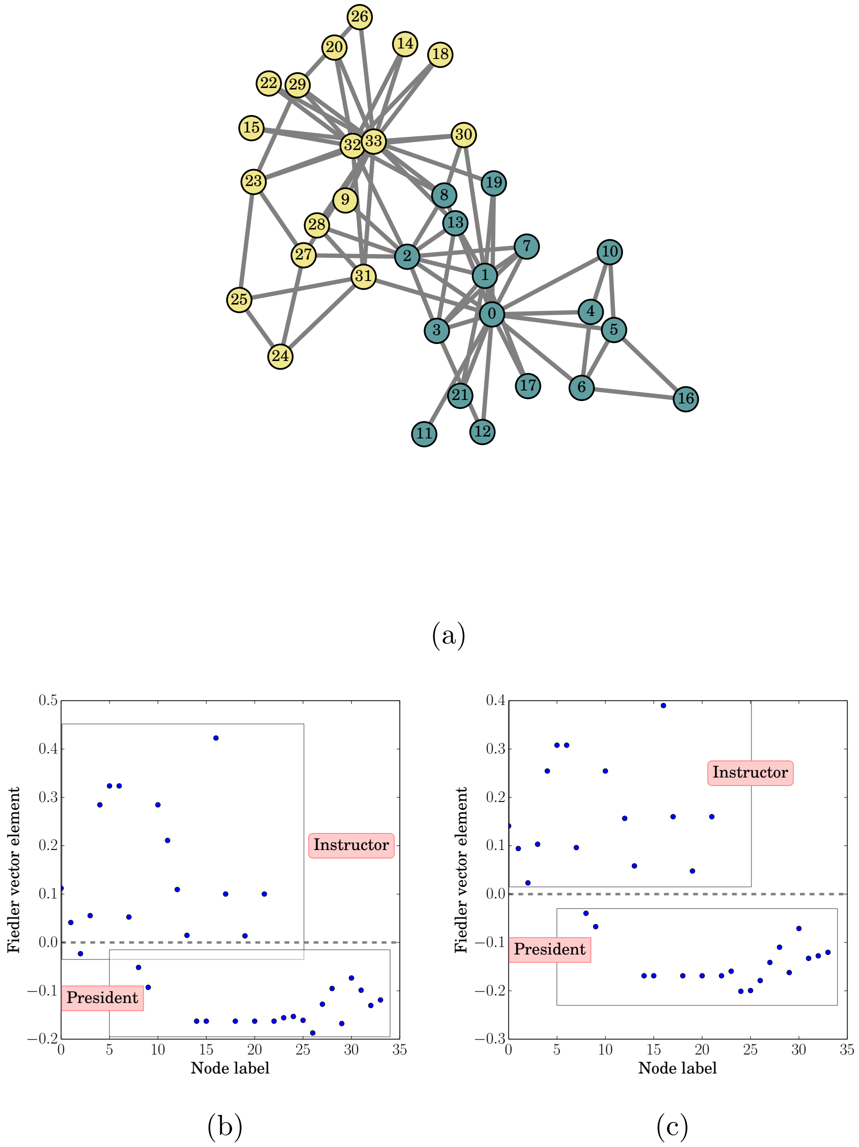
When a pair of nodes are in the same group, it is more likely to have strong flow of communication among each other together with their groupmates and information tends to stay within communities. This emphasizes the importance of having many non-overlapping short loops.

Analyzing loops in a network provides information about the cluster structure and emphasizes the importance of links in these clusters. By (5.4) the extremal density $\rho^*(e)$ measures the amount of important loops (see Section 5.1) passing through link e (expected usage). Assuming members of the community shares a lot of cycles between themselves, thus $\rho^*(e)$ serves as a measure of affinity for the nodes connected by e. In other words, nodes on important loops are well connected to the rest of the group. In this section, we show that indeed preprocessing the network using $\rho^*(e)$ can improve network partitioning.

After we compute loop modulus for a network, the extremal density $\rho^*(e)$ gives generic information about the structure of communities that contains many short loops and the importance of links in these clusters that generalize methods in[26] and[27]. We can substantially improve the performance of some partitioning methods such as spectral partitioning or modularity maximization heuristics by preprocessing the network into a weighted network with link weights $\rho^*(e)$. We can apply our methods to any weighted and directed network.

As the first example, we consider Zachary's Karate Club[17]–a friendships network at a university Karate club with 34 members, see Figure 5.11(a). A conflict between the instructor and the club's president split the club into two groups. Finding the communities in this network is a basic benchmark test for partitioning algorithms[119] Chapter 9.

To bisect this network, we use Fiedler vector bisection[67] on both weighted and unweighted networks in Figures 5.11(b) and (c). In the unweighted case, the bisection method failed to separate a node correctly and there are two nodes that are very close to the other cluster. Our weighting method does this clustering with complete accuracy.



(a)

(b)

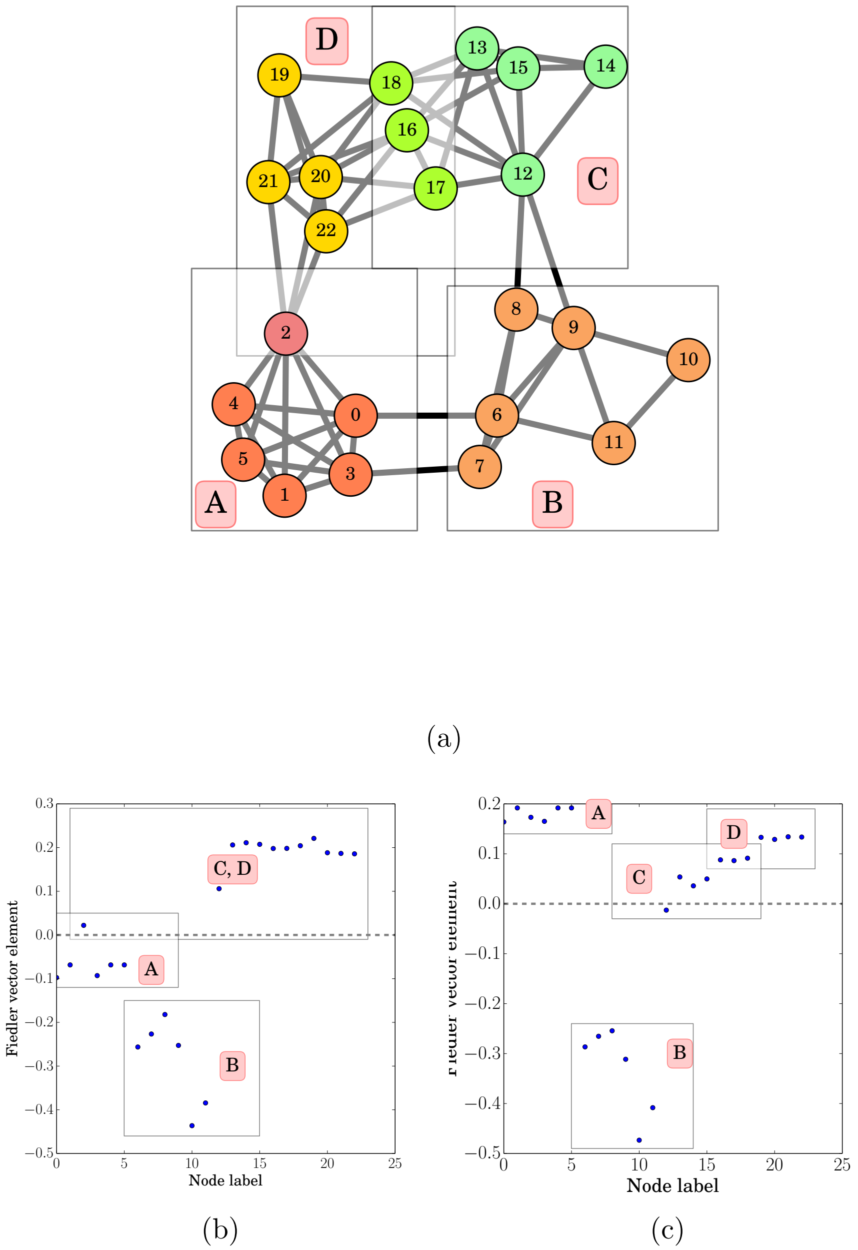
(c)

Figure 5.11: (a) Zachary's karate club network[17] with the groups splitted after conflict. (b)-(c) Fiedler vector values corresponding with the node labels. (b) Spectral partitioning of Zachary's karate club network[17], node 3 is wrongly partitioned. (c) spectral partitioning of the same network weighted by Loop Modulus where nodes are correctly partitioned.

It may be useful to allow for overlapping communities. For instance, a node can be a member of different communities, such as family, sport club, workplace, etc[120]. Although bisection methods alone are unable to detect overlapping communities, we see that loop modulus can augment these methods by distinguishing nested partitions in networks with overlapping communities in the next example. Figures 5.12 (a)–(c) show a network that is partitioned by Palla et al.[18]. We compute the Fiedler vector in both unweighted and weighted cases. As shown, the unweighted method failed to separate C and D overlapping communities, while the weighted method does distinguish them with the overlapping part.

To show the effectiveness of the weighting method in a more standard fashion, we consider two popular heuristics for modularity maximization; greedy modularity optimization method by Clauset, Newman, and Moore (CNM)[121] and the Louvain method[122] on the LFR benchmarks[123]. The LFR benchmarks allow the user to specify the community size distribution along with the degree distribution, offering more realistic benchmarks than the Girvan-Newman benchmarks[124]. We show re-weighting the network, using $\rho^*(e)$ from loop modulus, improve both CNM and Louvain substantially.

In Figure 5.13(a)-(c), three networks are produced by the LFR benchmark with 400 nodes, mean degree 5, maximum degree 10, and community sizes ranging from $20 - 40$ nodes. The interconnectedness of various communities is measured by the mixing rate μ. We plot the mutual information[125] for both the derived membership from CNM and Louvain on each network and the weighted version and compare them to the ground truth from LFR in Figure 5.13. As we observed, both the CNW and Louvain algorithms perform better on re-weighted networks using modulus.

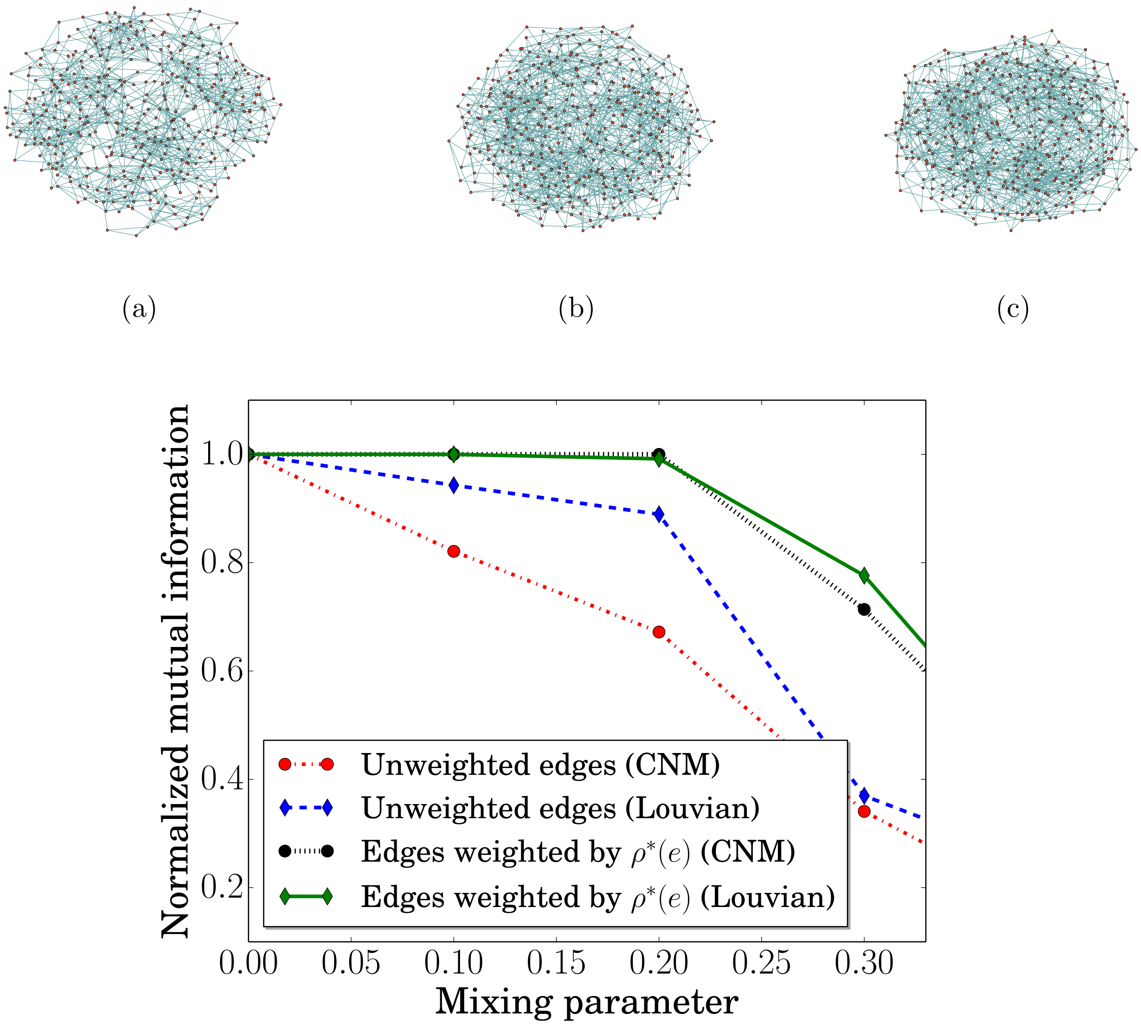


(a)

(b)

(c)

Figure 5.12: (a) A network partitioned by Palla et. al.[18]. Nodes 16, 17 and 18 are shared between C and D groups and Node 2 is shared between D and A groups. (b) Fiedler vector of the network, (c) Fiedler vector of the weighted network by Loop Modulus where overlapping groups can be distinguished.



(a) (b) (c)

(d)

Figure 5.13: (a)-(c) Networks are produced by LFR benchmark with size 400 nodes, mean degree 5, maximum degree 10, and community sizes ranging from $20 - 40$. The mixing rate μ, for adjusing ratio of intra-communities links over all links are 0.1, 0.2, and 0.3. (d) The plot depicts the normalized mutual information for community memberships found by Greedy modularity optimization (CNM) and Louvain method. Both the CNW and Louvain methods perform a better task on re-weighted networks.

Chapter 6

Conclusions and future work

6.1 Conclusions

In this dissertation, we framed modulus of families of walks as a tool for developing network measures that use generic structural properties of the network. We introduced general centrality measures based on modulus of families of walks. These measures provide information about nodes using knowledge from the entire network, while keeping computational costs low and without requiring acquisition of data from the entire network. These methods can be applied to very general networks, whether weighted, directed, multi-edged, or disconnected. We also presented several applications of our proposed measure to identify influential parts of a network and node ranking, as well as for mitigating epidemics. Considering different families of walks and their modulus can provide additional insights into solving other problems on networks.

We analyzed egocentric network measures based on modulus of family of walks connecting ego to its neighborhood nodes. We compare the proposed measures with the sociocentric counterparts and illustrate the advantages of our methods. For undirected networks, shell modulus can be computed by solving a Laplacian system. Moreover, for directed, multi-edges, networks we propose approximations that carry the same benefits of the original definition while being easy and cheap to compute. Finally, we introduce a generalization of degree called general degree. We illustrated the applications of our methods, particularly in epidemic mitigation.

We used modulus of families of loops to analyze loop structures in networks and showed that loop modulus quantifies the richness of loops in the network and we used it to measure clustering. The extremal densities found for loop modulus represent the probability of link participation in important loops. The performance of community detection methods such as spectral bisection and modularity maximization partitioning can be improved by weighting networks with their extremal densities derived from loop modulus.

We proposed a deterministic algorithm to find a shortest cycle in graphs. Instead of finding the shortest cycle rooted a each node, we focus on finding the shortest composite distance of a node to cycles in the graph. We proved that algorithm is $\mathcal{O}(\langle k\rangle n^2 \log n)$, and thus subcubic when graphs average degree is in $\mathcal{O}(n^{1-\epsilon})$, this is often the case in empirical applications.

6.2 Future work

Our research raised numerous questions that require further investigations. Here, we list a few of them briefly.

We focused primarily on edge-based modulus, where we assign a density to each edge, interpreted as the importance of the edge in a specific family. This can generalized to other network elements, such as nodes or sockets, leading to very different families of objects on networks.

Modulus characterizes the importance of members of a family from their dual point of view. For example, in a family of spanning trees, important members act as the backbone of the network. This is important for robustness and security design. Moreover, modulus sensitivity is determined by infinitesimal changes in the edge weights, using the extremal densities of the edges. Also, each member of the family has an associated dual variable, which provides a measure of its importance.

Moreover, analyzing network functions, such as synchronization and propagation, is another application. For example, investigating families of specific structures, e.g. loops, provide valuable information for inferring how the topology affects specific dynamics, e.g. synchronization.

There are many algorithmic challenges hidden in the modulus computation, for example in Section 5.3 in this dissertation. Moving toward distributed solver and exploiting graph structures as spatially distributed systems to capture couplings between optimization variables in the cost functional and linear constraints is another interesting direction. In the active set method for quadratic programming, the determination an efficient and general method for choosing active constraints using the dual problem gave us initial promising results and needs more in-depth analysis.

During the work on network robustness (Appendix A), we observed that modulus of family of walks connecting layers of network generalizes popular measures such as algebraic connectivity with fewer restrictions. This can be an immediate interesting line of research. Moreover, combining modulus with control theory is another promising application. For example, modulus of family of spanning trees with encrypted message delivery time is one application.

Chapter 5, shows that studying loop modulus improves community detection. A theoretical framework to re-formulate communities of networks based on cyclic structure looks more promising than existing methods based on the density of edges in communities. This requires a model-based approach in unsupervised statistical learning and stochastic block models.

Although we present some applications of loop modulus, analyzing loop structures on the network can expose information about various dynamics, e.g., synchronization and propagation[126–128]. Combining loop modulus and a measure such as the diameter can give us an insight about dynamics on the network such as consensus[126], synchronization of Kuramoto oscillators[127], and susceptible-infected-susceptible SIS epidemic model[128] where loops have a crucial role. Another application is to evaluate the complexity of the network. For example, considering trees as simple graphs, adding more links to them corresponds to more loops, and hence, higher complexity. Moreover, loop modulus satisfies the axioms of graph complexity in[129].

Modulus of families of walks is proved to be a useful tool for characterizing structure and functions of complex networks. We hope this dissertation further spreads the word and also helps users to appreciate modulus practicality.

Bibliography

[1] Heman Shakeri, Nathan Albin, Faryad Darabi Sahneh, Pietro Poggi-Corradini, and Caterina Scoglio. Maximizing algebraic connectivity in interconnected networks. *Physical Review E*, 93(3):030301, 2016.

[2] H. Shakeri, F. D. Darabi, V. Preciado, P. Poggi-Corradini, and C. Scoglio. Optimal Information Dissemination Strategy to Promote Preventive Behaviors in Multilayer Epidemic Networks. *Mathematical Biosciences and Engineering*, 12(3), June 2015. Preprint.

[3] Julian J McAuley and Jure Leskovec. Learning to discover social circles in ego networks. In *NIPS*, volume 2012, pages 548–56, 2012.

[4] D. S. Sade. Sociometrics of macaca mulatta i. linkages and cliques in grooming matrices. *Folia Primatologica*, 18(3-4):196–223, 1972.

[5] M Penrose. *Random Geometric Graphs*. Oxford scholarship online. Oxford University Press, 2003. ISBN 9780198506263.

[6] A Barabasi and R Albert. Emergence of scaling in random networks. *Science*, 286 (5439):509–512, 1999.

[7] U Brandes and D Fleischer. Centrality measures based on current flow. *STACS 2005, Lecture Notes in Computer Science*, 3404:533–544, 2005.

[8] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[9] Mathew Penrose. *Random geometric graphs*. Number 5. Oxford University Press, 2003.

[10] Alden S Klovdahl. Social networks and the spread of infectious diseases: the aids example. *Social science & medicine*, 21(11):1203–1216, 1985.

[11] Duncan J Watts and Steven H Strogatz. Collective dynamics of small-worldnetworks. *nature*, 393(6684):440–442, 1998.

[12] Marián Boguñá, Romualdo Pastor-Satorras, Albert Díaz-Guilera, and Alex Arenas. Models of social networks based on social distance attachment. *Physical review E*, 70 (5):056122, 2004.

[13] Amanda L Traud, Peter J Mucha, and Mason A Porter. Social structure of facebook networks. *Physica A: Statistical Mechanics and its Applications*, 391(16):4165–4180, 2012.

[14] Pablo M Gleiser and Leon Danon. Community structure in jazz. *Advances in complex systems*, 6(04):565–573, 2003.

[15] Roger Guimera, Leon Danon, Albert Diaz-Guilera, Francesc Giralt, and Alex Arenas. Self-similar community structure in a network of human interactions. *Physical review E*, 68(6):065103, 2003.

[16] Hamsterster friendships network dataset, KONECT. http://konect.uni-koblenz.de/networks/petster-friendships-hamster. Accessed: 2016-08-11.

[17] Wayne W Zachary. An information flow model for conflict and fission in small groups. *Journal of anthropological research*, pages 452–473, 1977.

[18] Gergely Palla, Imre Derényi, Illés Farkas, and Tamás Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435 (7043):814–818, 2005.

[19] Faryad Darabi Sahneh and Caterina M. Scoglio. Optimal information dissemination in epidemic networks. In *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, pages 1657–1662. IEEE, 2012.

[20] Phillip Schumm, Walter Schumm, and Caterina Scoglio. Impact of preventive responses to epidemics in rural regions. *PloS one*, 8(3):e59028, 2013.

[21] Duanbing Chen, Linyuan Lü, Ming-Sheng Shang, Yi-Cheng Zhang, and Tao Zhou. Identifying influential nodes in complex networks. *Physica a: Statistical mechanics and its applications*, 391(4):1777–1787, 2012.

[22] Xiaohang Zhang, Ji Zhu, Qi Wang, and Han Zhao. Identifying influential nodes in complex networks with community structure. *Knowledge-Based Systems*, 42:74–84, 2013.

[23] Amir Sheikhahmadi, Mohammad Ali Nematbakhsh, and Arman Shokrollahi. Improving detection of influential nodes in complex networks. *Physica A: Statistical Mechanics and its Applications*, 436:833–845, 2015.

[24] David C Bell, John S Atkinson, and Jerry W Carlson. Centrality measures for disease transmission networks. *Social networks*, 21(1):1–21, 1999.

[25] Mark EJ Newman. A measure of betweenness centrality based on random walks. *Social networks*, 27(1):39–54, 2005.

[26] Filippo Radicchi, Claudio Castellano, Federico Cecconi, Vittorio Loreto, and Domenico Parisi. Defining and identifying communities in networks. *Proceedings of the National Academy of Sciences of the United States of America*, 101(9):2658–2663, 2004.

[27] I Vragović and E Louis. Network community structure and loop coefficient method. *Physical Review E*, 74(1):016105, 2006.

[28] Santo Fortunato. Community detection in graphs. *Physics reports*, 486(3):75–174, 2010.

[29] Jonathan W Berry, Bruce Hendrickson, Randall A LaViolette, and Cynthia A Phillips. Tolerating the community detection resolution limit with edge weighting. *Physical Review E*, 83(5):056119, 2011.

[30] Santo Fortunato and Marc Barthelemy. Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, 104(1):36–41, 2007.

[31] Alireza Khadivi, Ali Ajdari Rad, and Martin Hasler. Network community-detection enhancement by proper weighting. *Physical Review E*, 83(4):046104, 2011.

[32] Linton C Freeman. A set of measures of centrality based on betweenness. *Sociometry*, pages 35–41, 1977.

[33] Stephen P Borgatti, Ajay Mehra, Daniel J Brass, and Giuseppe Labianca. Network analysis in the social sciences. *science*, 323(5916):892–895, 2009.

[34] Elizabeth M Daly and Mads Haahr. Social network analysis for routing in disconnected delay-tolerant manets. In *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*, pages 32–40. ACM, 2007.

[35] Stanley Wasserman and Katherine Faust. *Social network analysis: Methods and applications*, volume 8. Cambridge university press, 1994.

[36] Martin Everett and Stephen P Borgatti. Ego network betweenness. *Social networks*, 27(1):31–38, 2005.

[37] Shuai Gao, Jun Ma, Zhumin Chen, Guanghui Wang, and Changming Xing. Ranking the spreading ability of nodes in complex networks based on local structure. *Physica A: Statistical Mechanics and its Applications*, 403:130–147, 2014.

[38] Cai Gao, Daijun Wei, Yong Hu, Sankaran Mahadevan, and Yong Deng. A modified evidential methodology of identifying influential nodes in weighted networks. *Physica A: Statistical Mechanics and its Applications*, 392(21):5490–5500, 2013.

[39] Peter V Marsden. Egocentric and sociocentric measures of network centrality. *Social networks*, 24(4):407–422, 2002.

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

[41] Elizabeth Costenbader and Thomas W Valente. The stability of centrality measures when networks are sampled. *Social networks*, 25(4):283–307, 2003.

[42] Barbara Zemljič and Valentina Hlebec. Reliability of measures of centrality and prominence. *Social Networks*, 27(1):73–88, 2005.

[43] Ron Milo, Shai Shen-Orr, Shalev Itzkovitz, Nadav Kashtan, Dmitri Chklovskii, and Uri Alon. Network motifs: simple building blocks of complex networks. *Science*, 298 (5594):824–827, 2002.

[44] Salomon Mugisha and Hai-Jun Zhou. Identifying optimal targets of network attack by belief propagation. *Phys. Rev. E*, 94:012305, Jul 2016. doi: 10.1103/PhysRevE.94.012305. URL http://link.aps.org/doi/10.1103/PhysRevE.94.012305.

[45] Thomas Petermann and Paolo De Los Rios. Role of clustering and gridlike ordering in epidemic spreading. *Physical Review E*, 69(6):066116, 2004.

[46] Mark EJ Newman. The structure and function of complex networks. *SIAM review*, 45 (2):167–256, 2003.

[47] Pedro G Lind, Marta C González, and Hans J Herrmann. Cycles and clustering in bipartite networks. *Physical review E*, 72(5):056127, 2005.

[48] Ginestra Bianconi and Andrea Capocci. Number of loops of size h in growing scale-free networks. *Physical review letters*, 90(7):078701, 2003.

[49] Hyun-Joo Kim and Jin Min Kim. Cyclic topology in complex networks. *Physical Review E*, 72(3):036109, 2005.

[50] Nathan Albin, Pietro Poggi-Corradini, Faryad Darabi Sahneh, and Max Goering. Modulus of families of walks on graphs. In *Proceedings of Complex Analysis and Dynamical Systems VII*, to appear. http://arxiv.org/abs/1401.7640.

[51] Nathan Albin, Megan Brunner, Roberto Perez, Pietro Poggi-Corradini, and Natalie Wiens. Modulus on graphs as a generalization of standard graph theoretic quantities.

Conformal Geometry and Dynamics of the American Mathematical Society, 19(13): 298–317, 2015.

[52] Nathan Albin and Pietro Poggi-Corradini. Minimal subfamilies and the probabilistic interpretation for modulus on graphs. *The Journal of Analysis*, pages 1–26, 2016.

[53] L. V. Ahlfors. *Conformal invariants: topics in geometric function theory*. McGraw-Hill Book Co., New York, 1973. McGraw-Hill Series in Higher Mathematics.

[54] L. R. Ford and D. R. Fulkerson. Maximal flow through a network. *Canadian Journal of Mathematics 8*, 3:399–404, 1956.

[55] R. J. Duffin. The extremal length of a network. *J. Math. Anal. Appl.*, 5:200–215, 1962. ISSN 0022-247x.

[56] O Schramm. Square tilings with prescribed combinatorics. *Israel Journal of Mathematics*, 84(1-2):97–118, 1993.

[57] P. Haïssinsky. Empilements de cercles et modules combinatoires. *Annales de linstitut Fourier*, 59(6):2175–2222, 2009.

[58] J Ericson, P Poggi-Corradini, and H Zhang. Effective resistance on graphs and the epidemic quasimetric. *Involve, a Journal of Mathematics*, 2013.

[59] Heman Shakeri, Pietro Poggi-Corradini, Caterina Scoglio, and Nathan Albin. Generalized network measures based on modulus of families of walks. *Journal of Computational and Applied Mathematics*, 2016.

[60] Max Goering, Faryad Darabi Sahneh, Nathan Albin, Caterina Scoglio, and Pietro Poggi-Corradini. Numerical investigation of metrics for epidemic processes on graphs. *arXiv preprint arXiv:1511.07893*, 2015.

[61] Nathan Albin, Jason Clemens, and Pietro Poggi-Corradini. Blocking duality for p-modulus on networks. *arXiv preprint arXiv:1612.00435*, 2016.

[62] S. P. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

[63] Donald Goldfarb and Ashok Idnani. A numerically stable dual method for solving strictly convex quadratic programs. *Mathematical programming*, 27(1):1–33, 1983.

[64] G. Sabidussi. The centrality index of a graph. *Psychometrika*, 31(4):581–603, 1966. ISSN 0033-3123. doi: 10.1007/BF02289527. URL http://dx.doi.org/10.1007/BF02289527.

[65] A Landherr, B Friedl, and J Heidemann. A critical review of centrality measures in social networks. *Business and Information Systems Engineering*, 2:371–385, December 2010.

[66] L. C. Freeman. A set of measures of centrality -. *Sociometry*, 40(1):35–41, 1977.

[67] M. E. J. Newman. *Networks: An Introduction*. Oxford, 2010.

[68] T. Opsahl, F. Agneessensb, and J. Skvoretzc. Node centrality in weighted networks: Generalizing degree and shortest paths. *Social Networks*, 32:245–251, July 2010.

[69] S. C. Freeman and L. C. Freeman. The networkers network: A study of the impact of a new communications medium on sociometric structure. *Social Science Research Reports*, 1979.

[70] R. Cohen, S. Havlin, and D. ben Avraham. Efficient immunization strategies for computer networks and populations. *Phys. Rev. Lett.*, 91:247901, Dec 2003. doi: 10.1103/PhysRevLett.91.247901. URL http://link.aps.org/doi/10.1103/PhysRevLett.91.247901.

[71] Pratha Sah, Lisa O Singh, Aaron Clauset, and Shweta Bansal. Exploring community structure in biological networks with random graphs. *BMC bioinformatics*, 15(1):220, 2014.

[72] F. Darabi Sahneh, Heman Shakeri, Aram Vajdi, Futing Fan, and Caterina Scoglio. *GEMF: Generalized Epidemic Modelling Framework*. Network Science and Engineering Group (NETSE), Kansas State University, 1 edition, June 2015.

[73] Karen Stephenson and Marvin Zelen. Rethinking centrality: Methods and examples. *Social networks*, 11(1):1–37, 1989.

[74] Douglas J Klein and Milan Randić. Resistance distance. *Journal of mathematical chemistry*, 12(1):81–95, 1993.

[75] Heman Shakeri, Pietro Poggi-Corradini, Nathan Albin, and Caterina Scoglio. Network clustering and community detection using modulus of families of loops. *Phys. Rev. E*, 95:012316, Jan 2017. doi: 10.1103/PhysRevE.95.012316. URL https://link.aps.org/doi/10.1103/PhysRevE.95.012316.

[76] Juan Antonio Carrasco, Bernie Hogan, Barry Wellman, and Eric J Miller. Collecting social network data to study social activity-travel behavior: an egocentric approach. *Environment and Planning B: Planning and Design*, 35(6):961–980, 2008.

[77] Lars V. Ahlfors. *Conformal invariants: topics in geometric function theory*. McGraw-Hill Book Co., New York-Düsseldorf-Johannesburg, 1973. McGraw-Hill Series in Higher Mathematics.

[78] Russell Lyons and Yuval Peres. *Probability on trees and networks*, volume 42. Cambridge University Press, 2016.

[79] David Lusseau, Karsten Schneider, Oliver J Boisseau, Patti Haase, Elisabeth Slooten, and Steve M Dawson. The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. *Behavioral Ecology and Sociobiology*, 54 (4):396–405, 2003.

[80] Allison Davis, Burleigh Bradford Gardner, and Mary R Gardner. *Deep South: A social anthropological study of caste and class*. Univ of South Carolina Press, 2009.

[81] Romualdo Pastor-Satorras and Alessandro Vespignani. Immunization of complex networks. *Physical Review E*, 65(3):036104, 2002.

[82] Adilson E Motter and Ying-Cheng Lai. Cascade-based attacks on complex networks. *Physical Review E*, 66(6):065102, 2002.

[83] Ming Zhao, Tao Zhou, Bing-Hong Wang, and Wen-Xu Wang. Enhanced synchronizability by structural perturbations. *Physical Review E*, 72(5):057102, 2005.

[84] Yiping Chen, Gerald Paul, Shlomo Havlin, Fredrik Liljeros, and H Eugene Stanley. Finding a better immunization strategy. *Physical review letters*, 101(5):058701, 2008.

[85] Marcel Salathé and James H Jones. Dynamics and control of diseases in networks with community structure. *PLoS Comput Biol*, 6(4):e1000736, 2010.

[86] Reuven Cohen, Shlomo Havlin, and Daniel Ben-Avraham. Efficient immunization strategies for computer networks and populations. *Physical review letters*, 91(24): 247901, 2003.

[87] Faryad Darabi Sahneh, Aram Vajdi, Heman Shakeri, Futing Fan, and Caterina Scoglio. Gemfsim: a stochastic simulator for the generalized epidemic modeling framework. *arXiv preprint arXiv:1604.02175*, 2016.

[88] Henry B Mann and Donald R Whitney. On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, pages 50–60, 1947.

[89] Frank Harary et al. Graph theory, 1969.

[90] Petra M Gleiss, Josef Leydold, and Peter F Stadler. Circuit bases of strongly connected digraphs. 2001.

[91] Telikepalli Kavitha, Kurt Mehlhorn, Dimitrios Michail, and Katarzyna Paluch. A faster algorithm for minimum cycle basis of graphs. In *Automata, languages and programming*, pages 846–857. Springer, 2004.

[92] Telikepalli Kavitha, Kurt Mehlhorn, and Dimitrios Michail. New approximation algorithms for minimum cycle bases of graphs. In *STACS 2007*, pages 512–523. Springer, 2007.

[93] Alberto Caprara, Alessandro Panconesi, and Romeo Rizzi. Packing cycles in undirected graphs. *Journal of Algorithms*, 48(1):239–256, 2003.

[94] Michael Krivelevich, Zeev Nutov, and Raphael Yuster. Approximation algorithms for cycle packing problems. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 556–561. Society for Industrial and Applied Mathematics, 2005.

[95] Mohammad R Salavatipour and Jacques Verstraete. Disjoint cycles: Integrality gap, hardness, and approximation. In *Integer Programming and Combinatorial Optimization*, pages 51–65. Springer, 2005.

[96] Reinhard Diestel. *Graph theory* {*graduate texts in mathematics; 173*}. Springer-Verlag Berlin and Heidelberg GmbH & amp, 2000.

[97] Hristo N Djidjev. Computing the girth of a planar graph. In *Automata, Languages and Programming*, pages 821–831. Springer, 2000.

[98] Alon Itai and Michael Rodeh. Finding a minimum circuit in a graph. *SIAM Journal on Computing*, 7(4):413–423, 1978.

[99] Liam Roditty and Virginia Vassilevska Williams. Minimum weight cycles and triangles: Equivalences and algorithms. In *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*, pages 180–189. IEEE, 2011.

[100] Virginia Vassilevska Williams and Ryan Williams. Subcubic equivalences between path, matrix and triangle problems. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pages 645–654. IEEE, 2010.

[101] L. Roditty and R. Tov. Approximating the girth. pages 1446–1454, 2011. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-79955727144&partnerID=40&md5=9e02cbd736c40eb01240c9e982db5c5c. cited By 2.

[102] Andrzej Lingas and Eva-Marta Lundell. Efficient approximation algorithms for shortest cycles in undirected graphs. *Information Processing Letters*, 109(10):493–498, 2009.

[103] Raphael Yuster. A shortest cycle for each vertex of a graph. *Information Processing Letters*, 111(21):1057–1061, 2011.

[104] David Peleg, Liam Roditty, and Elad Tal. Distributed algorithms for network diameter and girth. *Automata, Languages, and Programming*, pages 660–672, 2012.

[105] Raphael Yuster and Uri Zwick. Finding even cycles even faster. *SIAM Journal on Discrete Mathematics*, 10(2):209–222, 1997.

[106] James B Orlin and Antonio Sedeno-Noda. An o (nm) time algorithm for finding the min length directed cycle in a graph, 2016.

[107] Thomas H Cormen. *Introduction to algorithms*. MIT press, 2009.

[108] Keith Paton. An algorithm for finding a fundamental set of cycles of a graph. *Communications of the ACM*, 12(9):514–518, 1969.

[109] Mark EJ Newman. Ego-centered networks and the ripple effect. *Social Networks*, 25 (1):83–95, 2003.

[110] Guido Caldarelli, Romualdo Pastor-Satorras, and Alessandro Vespignani. Structure of cycles and local ordering in complex networks. *The European Physical Journal B-Condensed Matter and Complex Systems*, 38(2):183–186, 2004.

[111] Pedro G Lind and Hans J Herrmann. New approaches to model and study social networks. *New Journal of Physics*, 9(7):228, 2007.

[112] Agata Fronczak, Janusz A Hołyst, Maciej Jedynak, and Julian Sienkiewicz. Higher order clustering coefficients in barabási–albert networks. *Physica A: Statistical Mechanics and its Applications*, 316(1):688–694, 2002.

[113] Sara Nadiv Soffer and Alexei Vazquez. Network clustering coefficient without degree-correlation biases. *Physical Review E*, 71(5):057101, 2005.

[114] Jari Saramäki, Mikko Kivelä, Jukka-Pekka Onnela, Kimmo Kaski, and Janos Kertesz. Generalizations of the clustering coefficient to weighted complex networks. *Physical Review E*, 75(2):027105, 2007.

[115] Tore Opsahl and Pietro Panzarasa. Clustering in weighted networks. *Social networks*, 31(2):155–163, 2009.

[116] Jérôme Kunegis. Handbook of network analysis [konect–the koblenz network collection]. *arXiv preprint arXiv:1402.5500*, 2014.

[117] George C Homans. *The human group*, volume 7. Routledge, 2013.

[118] Erzsébet Ravasz, Anna Lisa Somera, Dale A Mongru, Zoltán N Oltvai, and A-L Barabási. Hierarchical organization of modularity in metabolic networks. *science*, 297(5586):1551–1555, 2002.

[119] Albert-László Barabási. Network science. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 371(1987): 20120375, 2013.

[120] Gergely Palla, Albert-László Barabási, and Tamás Vicsek. Quantifying social group evolution. *Nature*, 446(7136):664–667, 2007.

[121] Aaron Clauset, Mark EJ Newman, and Cristopher Moore. Finding community structure in very large networks. *Physical review E*, 70(6):066111, 2004.

[122] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

[123] Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi. Benchmark graphs for testing community detection algorithms. *Physical review E*, 78(4):046110, 2008.

[124] Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826, 2002.

[125] Leon Danon, Albert Diaz-Guilera, Jordi Duch, and Alex Arenas. Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(09):P09008, 2005.

[126] Zhongkui Li, Zhisheng Duan, Guanrong Chen, and Lin Huang. Consensus of multiagent systems and synchronization of complex networks: a unified viewpoint. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 57(1):213–224, 2010.

[127] Yoshiki Kuramoto. Self-entrainment of a population of coupled non-linear oscillators. In *International symposium on mathematical problems in theoretical physics*, pages 420–422. Springer, 1975.

[128] Piet Van Mieghem. The n-intertwined sis epidemic network model. *Computing*, 93 (2-4):147–169, 2011.

[129] Carter T Butts. An axiomatic approach to network complexity. *Journal of Mathematical Sociology*, 24(4):273–301, 2000.

[130] Mikko Kivelä, Alex Arenas, Marc Barthelemy, James P Gleeson, Yamir Moreno, and Mason A Porter. Multilayer networks. *Journal of Complex Networks*, 2(3):203–271, 2014.

[131] Jean-Claude Laprie, Karama Kanoun, and Mohamed Kaâniche. Modelling interdependencies between the electricity and information infrastructures. In *Computer Safety, Reliability, and Security*, pages 54–67. Springer, 2007.

[132] Stefano Panzieri and Roberto Setola. Failures propagation in critical interdependent infrastructures. *International Journal of Modelling, Identification and Control*, 3(1): 69–78, 2008.

[133] Sergey V Buldyrev, Roni Parshani, Gerald Paul, H Eugene Stanley, and Shlomo Havlin. Catastrophic cascade of failures in interdependent networks. *Nature*, 464(7291):1025–1028, 2010.

[134] A Jamakovic and S Uhlig. On the relationship between the algebraic connectivity and graph's robustness to node and link failures. In *Next Generation Internet Networks, 3rd EuroNGI Conference on*, pages 96–102. IEEE, 2007.

[135] Miroslav Fiedler. Algebraic connectivity of graphs. *Czechoslovak mathematical journal*, 23(2):298–305, 1973.

[136] Shaun M Fallat, Steve Kirkland, and Sukanta Pati. On graphs with algebraic connectivity equal to minimum edge density. *Linear algebra and its applications*, 373:31–50, 2003.

[137] Pierre Brémaud. *Markov chains: Gibbs fields, Monte Carlo simulation, and queues*, volume 31. Springer Science & Business Media, 2013.

[138] Stephen Boyd, Persi Diaconis, and Lin Xiao. Fastest mixing markov chain on a graph. *SIAM review*, 46(4):667–689, 2004.

[139] Jun Sun, Stephen Boyd, Lin Xiao, and Persi Diaconis. The fastest mixing markov process on a graph and a connection to a maximum variance unfolding problem. *SIAM review*, 48(4):681–699, 2006.

[140] Sergio Gomez, Albert Diaz-Guilera, Jesus Gomez-Gardeñes, Conrad J Perez-Vicente, Yamir Moreno, and Alex Arenas. Diffusion dynamics on multiplex networks. *Physical review letters*, 110(2):028701, 2013.

[141] Albert Sole-Ribalta, Manlio De Domenico, Nikos E Kouvaris, Albert Diaz-Guilera, Sergio Gomez, and Alex Arenas. Spectral properties of the laplacian of multiplex networks. *Physical Review E*, 88(3):032807, 2013.

[142] Filippo Radicchi and Alex Arenas. Abrupt transition in the structural formation of interconnected networks. *Nature Physics*, 9(11):717–720, 2013.

[143] Faryad Darabi Sahneh, Caterina Scoglio, and Piet Van Mieghem. Exact coupling threshold for structural transition reveals diversified behaviors in interconnected networks. *Phys. Rev. E*, 92:040801, Oct 2015. doi: 10.1103/PhysRevE.92.040801. URL http://link.aps.org/doi/10.1103/PhysRevE.92.040801.

[144] J Martín-Hernández, H Wang, P Van Mieghem, and G DAgostino. Algebraic connectivity of interdependent networks. *Physica A: Statistical Mechanics and its Applications*, 404:92–105, 2014.

[145] Xin Li, Haotian Wu, Caterina Scoglio, and Don Gruenbacher. Robust allocation of weighted dependency links in cyber–physical networks. *Physica A: Statistical Mechanics and its Applications*, 433:316–327, 2015.

[146] Cyrus D Cantrell. *Modern mathematical methods for physicists and engineers*. Cambridge University Press, 2000.

[147] Faryad Darabi Sahneh, Fahmida N. Chowdhury, and Caterina M. Scoglio. On the existence of a threshold for preventive behavioral responses to suppress epidemic spreading. *Sci. Rep.*, 2:–, September 2012. URL http://dx.doi.org/10.1038/srep00632.

[148] Faryad Darabi Sahneh and Caterina Scoglio. Epidemic spread in human networks. In *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, pages 3008–3013. IEEE, 2011.

[149] Gui-Quan Sun, Quan-Xing Liu, Zhen Jin, Amit Chakraborty, and Bai-Lian Li. Influence of infection rate and migration on extinction of disease in spatial epidemics. *Journal of Theoretical Biology*, 264(1):95–103, 2010.

[150] Gui-Quan Sun. Pattern formation of an epidemic model with diffusion. *Nonlinear Dynamics*, 69(3):1097–1104, 2012.

[151] Clara Granell, Sergio Gómez, and Alex Arenas. Dynamical interplay between awareness and epidemic spreading in multiplex networks. *Physical review letters*, 111(12): 128701, 2013.

[152] Sebastian Funk and Vincent AA Jansen. Interacting epidemics on overlay networks. *Physical Review E*, 81(3):036118, 2010.

[153] Mark Dickison, Shlomo Havlin, and H Eugene Stanley. Epidemics on interconnected networks. *Physical Review E*, 85(6):066109, 2012.

[154] Anna Saumell-Mendiola, M Ángeles Serrano, and Marián Boguná. Epidemic spreading on interconnected networks. *Physical Review E*, 86(2):026106, 2012.

[155] Osman Yağan and Virgil Gligor. Analysis of complex contagions in random multiplex networks. *Physical Review E*, 86(3):036103, 2012.

[156] Ali Tavasoli, Mahyar Naraghi, and Heman Shakeri. Optimized coordination of brakes and active steering for a 4ws passenger car. *ISA transactions*, 51(5):573–583, 2012.

[157] Stefano Boccaletti, Ginestra Bianconi, Regino Criado, Charo I Del Genio, Jesús Gómez-Gardenes, Miguel Romance, Irene Sendina-Nadal, Zhen Wang, and Massimiliano Zanin. The structure and dynamics of multilayer networks. *Physics Reports*, 544(1):1–122, 2014.

[158] Victor M. Preciado, Faryad Darabi Sahneh, and Caterina Scoglio. A convex framework for optimal investment on disease awareness in social networks. In *Global Conference on Signal and Information Processing (GlobalSIP), 2013 IEEE*, pages 851–854, 2013.

[159] B Lemmens and R Nussbaum. *Nonlinear Perron-Frobenius Theory*. Cambridge Tracts in Mathematics. Cambridge University Press, 2012. ISBN 9780521898812. URL http://books.google.com/books?id=EYud2hfi_c4C.

[160] A. Charnes and W. W. Cooper. Programming with linear fractional functionals. *Naval Research Logistics*, 9:181?186, 1962.

[161] Michael Grant, Stephen Boyd, and Yinyu Ye. Cvx: Matlab software for disciplined convex programming, 2008.

[162] F Darabi Sahneh, Caterina Scoglio, and Piet Van Mieghem. Generalized epidemic mean-field model for spreading processes over multilayer complex networks. *Networking, IEEE/ACM Transactions on*, 21(5):1609–1620, 2013.

[163] Daniel A Schult and P Swart. Exploring network structure, dynamics, and function using networkx. In *Proceedings of the 7th Python in Science Conferences (SciPy 2008)*, volume 2008, pages 11–16, 2008.

[164] Zhen Z Shi, Chih-Hang Wu, and David Ben-Arieh. Agent-based model: a surging tool to simulate infectious diseases in the immune system. *Open Journal of Modelling and Simulation*, 2014, 2014.

[165] Chih-Hang J Wu, ZhenZhen Shi, David Ben-Arieh, Steven Q Simpson, and Douglas Peterson. Agent-based model with embedded system dynamics: A simulation tool for modeling progression of acute inflammatory responses. In *D37. IMMUNE MECHANISMS IN THE LUNG*, pages A5729–A5729. American Thoracic Society, 2010.

[166] Zhenzhen Shi, Stephen K Chapes, David Ben-Arieh, and Chih-Hang Wu. An agent-based model of a hepatic inflammatory response to salmonella: A computational study under a large set of experimental data. *PloS one*, 11(8):e0161131, 2016.

[167] Faryad Darabi Sahneh and Caterina Scoglio. Competitive epidemic spreading over arbitrary multilayer networks. *Physical Review E*, 89(6):062817, 2014.

[168] Daniel T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *Physical Chemistry*, 1977.

[169] M.J. Keeling and P. Rohani. *Modeling Infectious Diseases in Humans and Animals*. Princeton University Press, 2007.

Appendix A

Maximizing algebraic connectivity in interconnected networks[1]

Algebraic connectivity, the second eigenvalue of the Laplacian matrix, is a measure of node and link connectivity on networks. When studying interconnected networks it is useful to consider a multiplex model, where the component networks operate together with inter-layer links among them. In order to have a well-connected multilayer structure, it is necessary to optimally design these inter-layer links considering realistic constraints. In this work, we solve the problem of finding an optimal weight distribution for one-to-one inter-layer links under budget constraint. We show that for the special multiplex configurations with identical layers, the uniform weight distribution is always optimal. On the other hand, when the two layers are arbitrary, increasing the budget reveals the existence of two different regimes. Up to a certain threshold budget, the second eigenvalue of the supra-Laplacian is simple, the optimal weight distribution is uniform, and the Fiedler vector is constant on each layer. Increasing the budget past the threshold, the optimal weight distribution can be non-uniform. The interesting consequence of this result is that there is no need to solve the optimization problem when the available budget is less than the threshold, which can be easily found analytically.

Real-world networks are often connected together and therefore influence each other[130].

Robust design of interdependent networks is critical to allow uninterrupted flow of information, power, and goods in spite of possible errors and attacks[131–133]. The second eigenvalue of the Laplacian matrix, $\lambda_2(L)$, is a good measure of network robustness[134]. Fiedler shows that algebraic connectivity increases by adding links[135]. Moreover, it is harder to bisect a network with higher algebraic connectivity[136].

The second eigenvalue of the Laplacian matrix is also a measure of the speed of mixing for a Markov process on a network[137]. Boyd et al. maximize the mixing rate by assigning optimum link weights in the setting of a single layer[138;139].

For multiplex networks (see Fig. A.1), a natural question is the following. Given fixed network layers, how should the weights be assigned to inter-layer links in order to maximize algebraic connectivity?

The behavior of λ_2, in the case of identical weights, i.e., with a fixed coupling weight p for every inter-layer link, has been studied recently. For instance, Gomez et al. observe that $\lambda_2(L)$ grows linearly with p up to a critical p^*, and then has a non-linear behavior afterwards[140]. Sole-Ribalta et al. analyze the spectrum of multiplex networks with perturbation theory on a decomposed–the intra- and interlayer structure–version of Laplacian matrix[141].

Radicchi and Arenas find bounds for this threshold value p^*[142]. Sahneh et al. compute the exact value analytically[143].

Martin-Hernandez et al. analyze the algebraic connectivity and Fiedler vector of multiplex structures, with addition of a number of inter-layer links in two configurations; diagonal (one-to-one) and random[144]. They show that for the first case, algebraic connectivity satu-

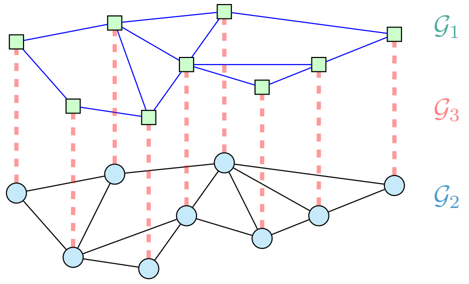


Figure A.1: A schematic of a multiplex network $\mathcal{G}$ with two layers $\mathcal{G}_1$, $\mathcal{G}_2$, connecting through an inter-layer one-to-one structure $\mathcal{G}_3$.

rates after adding a sufficient number of links. Li et al. adopt a network flow approach to propose a heuristic that improves robustness of large multiplex networks by choosing from a set of inter-layer links with predefined weights[145].

Here, we remove the constraint of identical interlinking weights and pose the problem of finding the maximum algebraic connectivity for a one-to-one interconnected structure between different layers in the presence of limited resources. We show that up to the threshold budget p^*N—where p^* is the same threshold studied before[140;142;143]—the uniform distribution of identical weights is actually optimal. For larger budgets, the optimal distribution of weights is generally not uniform.

A.1 Model framework

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ represents a network and by $\mathcal{V} = \{1, \dots, N\}$ and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$, we denote the set of nodes and links. For a link e between nodes u and v, i.e, $e : \{u, v\} \in \mathcal{E}$, we define a nonnegative value w_{uv} as the weight of the link. The Laplacian matrix of $\mathcal{G}$ can be defined as:

$$L = \sum_{ij \in \mathcal{E}} w_{ij} B_{ij} \tag{A.1}$$

where $B_{ij} := (e_i - e_j)(e_i - e_j)^T$ is the incidence matrix for link ij, and e_i is a vector with ith component one and rest of its elements are zero.

For a multiplex network with two layers $\mathcal{G}_1 = \{\mathcal{V}_1, \mathcal{E}_1\}$ and $\mathcal{G}_2 = \{\mathcal{V}_2, \mathcal{E}_2\}$ and $|\mathcal{V}_1| = |\mathcal{V}_2|$, we consider a bipartite graph $\mathcal{G}_3 = \{\mathcal{V}, \mathcal{E}_3\}$ with $\mathcal{E}_3 \subseteq \mathcal{V}_1 \times \mathcal{V}_2$. The multiplex network $\mathcal{G}$ is composed from $\mathcal{G}_1$, $\mathcal{G}_2$, and $\mathcal{G}_3$ (Fig. A.1). We want to design optimal weights for $\mathcal{G}_3$ to improve the algebraic connectivity of $\mathcal{G}$ as much as possible with a limited budget, i.e., $\sum w_{ij} = c$. Using Eq. (A.1), the Laplacian matrix of $\mathcal{G}$ (supra-Laplacian matrix), is:

$$L(w) = \sum_{ij \in \mathcal{E}_2 \cup \mathcal{E}_3} B_{ij} + \sum_{ij \in \mathcal{E}_3} w_{ij} B_{ij}, \tag{A.2}$$

where we use the notation $L(w)$ to make explicit the dependence of the Laplacian on the interlayer weights w.

From Eq. (A.2), the Laplacian, L, of the combined network takes the form

$$L(w) = \begin{bmatrix} L_1 & 0 \\ 0 & L_2 \end{bmatrix} + \begin{bmatrix} W & -W \\ -W & W \end{bmatrix},$$

where L_1 and L_2 are the Laplacians of the individual layers and $W = \mathrm{diag}(w)$ with $w \geq 0$ the inter-layer link weights satisfying the budget constraint $w^T\mathbf{1} = c$. We assume the two layers are connected independently, so that $\lambda_3(L) > 0$, for all choices of c and w.

The second eigenvalue can be characterized as the solution to the optimization problem

$$\lambda_2(L) = \min_{\substack{v \neq 0 \\ v^T\mathbf{1}=0}} \frac{v^T L v}{\|v\|^2}. \tag{A.3}$$

The optimal weight problem, then, can be phrased as follows. Given a budget $c \geq 0$, solve the problem

$$F(c) := \max_{\substack{w \geq 0 \\ w^T\mathbf{1}=c}} \lambda_2(L(w)). \tag{A.4}$$

Since L is an affine function of w, and λ_2 is a concave function of L, it follows that (A.4) is a convex optimization problem. In fact, it can be recast as a semi-definite programming problem or SDP:

$$\begin{aligned} \underset{w_{ij}}{\text{maximize}} \quad & \lambda \\ \text{subject to} \quad & \sum_{ij \in \mathcal{E}_3} w_{ij} B_{ij} + L_0 + \mu e e^T - \lambda I_n \succeq 0 \\ & \sum_{ij \in \mathcal{E}_3} w_{ij} \leq c \\ & w_{ij} \geq 0 \end{aligned} \tag{A.5}$$

where $L_0 = \sum_{i,j \in \mathcal{E}_1 \cup \mathcal{E}_2} B_{ij}$. We know $L \succeq 0$ and $\lambda_1 = 0$. Due to this redundancy in Laplacian matrix, parameter μ is employed to avoid the zero eigenvalue.

Problem (A.5) is a convex SDP[62] and can be efficiently solved for arbitrary large networks with applying sub-gradient methods. We consider the case of a two-layer network with one-to-one interlayer links.

A.2 Threshold for optimal weight distribution

Returning to (A.3), it is convenient to write v in component form $v = (v_1^T, v_2^T)^T$ so that (A.3) implies

$$\begin{aligned} v_1^T L_1 v_1 + & v_2^T L_2 v_2 + (v_1 - v_2)^T W (v_1 - v_2) \\ & - \lambda_2(L) \left(\|v_1\|^2 + \|v_2\|^2 \right) \geq 0 \qquad \forall\ v_1^T \mathbf{1} = -v_2^T \mathbf{1}. \end{aligned} \tag{A.6}$$

Since v must satisfy $v_1^T \mathbf{1} = -v_2^T \mathbf{1}$, we use the following substitution for v_1 and v_2 to separate the $\mathbf{1}$ subspace and its orthogonal counterpart u_i:

$$v_1 = \alpha \mathbf{1} + u_1, \qquad v_2 = -\alpha \mathbf{1} + u_2, \tag{A.7}$$

where $u_i \in \mathbb{R}^N$, such that $u_1^T \mathbf{1} = u_2^T \mathbf{1} = 0$, and α is some constant. Rewriting the terms in (A.6), we observe that

$$\begin{aligned} (v_1 - v_2)^T & W (v_1 - v_2) \\ & = (2\alpha \mathbf{1} + u_1 - u_2)^T W (2\alpha \mathbf{1} + u_1 - u_2) \\ & = 4\alpha^2 c + 4\alpha w^T (u_1 - u_2) \\ & \ + (u_1 - u_2)^T W (u_1 - u_2) \end{aligned}$$

and that

$$\|v_i\|^2 = \|\alpha \mathbf{1}\|^2 + \|u_i\|^2 = \alpha^2 N + \|u_i\|^2 \qquad \text{for } i = 1, 2.$$

Thus, Eq. (A.6) implies that

$$\begin{aligned} & u_1^T L_1 u_1 + u_2^T L_2 u_2 + 4\alpha^2 c + 4\alpha w^T (u_1 - u_2) \\ & + (u_1 - u_2)^T W (u_1 - u_2) - \\ & \lambda_2(L)\left(2\alpha^2 N + \|u_1\|^2 + \|u_2\|^2\right) \geq 0 \\ & \forall \alpha, u_1^T \mathbf{1} = u_2^T \mathbf{1} = 0. \end{aligned} \tag{A.8}$$

In particular, setting $u_1 = u_2 = 0$ in (A.8), then, gives the inequality

$$4\alpha^2 c - 2\alpha^2 N \lambda_2(L) \geq 0 \qquad \forall \alpha$$

which can only be true if $\lambda_2(L) \leq \frac{2c}{N}$. Thus for the two-layer problem described above, we have the bound

$$F(c) \leq \frac{2c}{N}. \tag{A.9}$$

Now we turn our attention to the question of attainability of (A.9). This question is answered by the following theorem.

Theorem A.2.1. *The inequality in* (A.9) *can only be satisfied as equality if* $w = \frac{c}{N}\mathbf{1}$.

Proof. Suppose the weights w are chosen such that the Laplacian L satisfies $\lambda_2(L) = \frac{2c}{N}$. Then (A.8) simplifies to

$$\begin{aligned} & u_1^T L_1 u_1 + u_2^T L_2 u_2 + 4\alpha w^T (u_1 - u_2) \\ & + (u_1 - u_2)^T W (u_1 - u_2) \\ & - \frac{2c}{N}\left(\|u_1\|^2 + \|u_2\|^2\right) \geq 0 \ \ \forall \ \alpha, u_1^T \mathbf{1} = u_2^T \mathbf{1} = 0. \end{aligned}$$

This can only be true if the linear coefficient in α, $4w^T(u_1 - u_2)$, vanishes for every choice of u_1, u_2 satisfying $u_1^T \mathbf{1} = u_2^T \mathbf{1} = 0$. This implies that w is parallel to $\mathbf{1}$ and, since $w^T \mathbf{1} = c$, the theorem follows. □

The previous theorem shows that when the bound (A.9) is attained, it can only be attained by the uniform choice of weights $w = \frac{c}{N}\mathbf{1}$. The next theorem characterizes exactly the budgets for which the bound is attained.

Theorem A.2.2. *For a given two-layer network, define the constant*

$$c^* := N \min_{\substack{u_1^T\mathbf{1}=u_2^T\mathbf{1}=0 \\ u_1+u_2\neq 0}} \frac{u_1^T L_1 u_1 + u_2^T L_2 u_2}{\|u_1+u_2\|^2} \tag{A.10}$$

Then, for all budgets $c \geq 0$, $F(c) = \frac{2c}{N}$ if and only if $c \leq c^*$.

Proof. By Theorem A.2.1, the upper-bound $\frac{2c}{N}$ for $F(c)$ can be attained only in the case of uniform weights $w = \frac{c}{N}\mathbf{1}$. In this case we write $L = L(c)$. For all $c \geq 0$, one can check that $\frac{2c}{N}$ is always an eigenvalue of $L(c)$, with eigenvector $(\mathbf{1}^T, -\mathbf{1}^T)^T$. Since $L(c)$ is positive semi-definite and $\lambda_1(L(c)) = 0$, it follows that $\lambda_2(L(c)) \leq \frac{2c}{N}$. Thus, we have $F(c) = \frac{2c}{N}$ if and only if $\lambda_2(L(c)) \geq \frac{2c}{N}$. Recalling the variational characterization of $\lambda_2(c)$ in (A.3), we observe that $\lambda_2(L(c)) \geq \frac{2c}{N}$ if and only if the following inequality holds for every choice of $v \neq 0$, with $v^T\mathbf{1} = 0$ or, equivalently, for every choice of α, u_1 and u_2 according to the substitution (A.7):

$$\begin{aligned} 0 &\leq v^T L v - \frac{2c}{N}\|v\|^2 \\ &= v_1^T L_1 v_1 + v_2^T L_2 v_2 + \frac{c}{N}\|v_1 - v_2\|^2 - \frac{2c}{N}\left(\|v_1\|^2 + \|v_2\|^2\right) \\ &= u_1^T L_1 u_1 + u_2^T L_2 u_2 - \frac{c}{N}\|u_1 + u_2\|^2. \end{aligned}$$

This inequality holds for all $u_1^T\mathbf{1} = u_2^T\mathbf{1} = 0$ if and only if $c \leq c^*$ as defined in (A.10), completing the proof.

□

The threshold obtained by Eq. (A.10) is exactly equivalent to the threshold found in[143], as shown in the following theorem.

Theorem A.2.3. *The threshold budget c^* satisfies*

$$\frac{c^*}{N} = \lambda_2\left(\left(L_1^\dagger + L_2^\dagger\right)^\dagger\right) \tag{A.11}$$

Proof. We begin by rewriting the minimization in (A.10):

$$\begin{aligned}\min_{\substack{u_1^T\mathbf{1}=u_2^T\mathbf{1}=0\\ u_1+u_2\neq 0}} \frac{u_1^T L_1 u_1 + u_2^T L_2 u_2}{\|u_1+u_2\|^2} &= \min_{\substack{u^T\mathbf{1}=0\\ u\neq 0}} \min_{\substack{u_1^T\mathbf{1}=u_2^T\mathbf{1}=0\\ u_1+u_2=u}} \frac{u_1^T L_1 u_1 + u_2^T L_2 u_2}{\|u\|^2}\\ &= \min_{\substack{u^T\mathbf{1}=0\\ u\neq 0}} \frac{1}{\|u\|^2} \min_{\substack{u_1^T\mathbf{1}=u_2^T\mathbf{1}=0\\ u_1+u_2=u}} \left(u_1^T L_1 u_1 + u_2^T L_2 u_2\right).\end{aligned} \tag{A.12}$$

To solve the inner minimization problems, we introduce Lagrange multipliers to find that the minimizing u_1 and u_2 satisfy

$$L_1 u_1 = \nu\mathbf{1} + \mu, \qquad L_2 u_2 = \eta\mathbf{1} + \mu.$$

Taking an inner product of each of these with the $\mathbf{1}$ vector shows that

$$\nu = \eta = -\frac{\mu^T\mathbf{1}}{N},$$

so that

$$u_1 = L_1^\dagger\left(\mu - \frac{\mu^T\mathbf{1}}{N}\right), \qquad u_2 = L_2^\dagger\left(\mu - \frac{\mu^T\mathbf{1}}{N}\right).$$

Thus, without loss of generality, μ can be taken to be orthogonal to $\mathbf{1}$. With this form, u_1 and u_2 are already orthongal to $\mathbf{1}$ as well. In order to satisfy the constraint $u_1 + u_2 = u$, we must have

$$\left(L_1^\dagger + L_2^\dagger\right)\mu = u, \quad \text{i.e.,} \quad \mu = \left(L_1^\dagger + L_2^\dagger\right)^\dagger u.$$

From this, we see that the minimizing u_1 and u_2 of the inner minimization problem in (A.12) satisfy

$$u_1 = L_1^\dagger\left(L_1^\dagger + L_2^\dagger\right)^\dagger u, \qquad u_2 = L_2^\dagger\left(L_1^\dagger + L_2^\dagger\right)^\dagger u,$$

giving a minimum value of

$$
\begin{aligned}
u_1^T L_1 u_1 + u_2^T L_2 u_2 &= u^T \left(L_1^\dagger + L_2^\dagger\right)^\dagger L_1^\dagger L_1 L_1^\dagger \left(L_1^\dagger + L_2^\dagger\right)^\dagger u + \\
&\qquad u^T \left(L_1^\dagger + L_2^\dagger\right)^\dagger L_2^\dagger L_2 L_2^\dagger \left(L_1^\dagger + L_2^\dagger\right)^\dagger u \\
&= u^T \left(L_1^\dagger + L_2^\dagger\right)^\dagger L_1^\dagger \left(L_1^\dagger + L_2^\dagger\right)^\dagger u + \\
&\qquad u^T \left(L_1^\dagger + L_2^\dagger\right)^\dagger L_2^\dagger \left(L_1^\dagger + L_2^\dagger\right)^\dagger u \\
&= u^T \left(L_1^\dagger + L_2^\dagger\right)^\dagger \left(L_1^\dagger + L_2^\dagger\right) \left(L_1^\dagger + L_2^\dagger\right)^\dagger u \\
&= u^T \left(L_1^\dagger + L_2^\dagger\right)^\dagger u.
\end{aligned}
\tag{A.13}
$$

Here, we have used the identity $A^\dagger A A^\dagger = A^\dagger$.

Substituting back into (A.12), we have

$$
\frac{c^*}{N} = \min_{\substack{u^T \mathbf{1} = 0 \\ u \neq 0}} \frac{u^T \left(L_1^\dagger + L_2^\dagger\right)^\dagger u}{\|u\|^2}.
$$

Since L_1 and L_2 are positive semidefinite, so are $L_1^\dagger$ and $L_2^\dagger$ and, consequently, so are $L_1^\dagger + L_2^\dagger$ and $\left(L_1^\dagger + L_2^\dagger\right)^\dagger$. Since the component networks are assumed connected, the nullspace of $\left(L_1^\dagger + L_2^\dagger\right)^\dagger$ is spanned by the vector $\mathbf{1}$. The Rayleigh quotient in (A.13) is therefore minimized over the orthogonal complement of the eigenspace associated with the first eigenvalue of $\left(L_1^\dagger + L_2^\dagger\right)^\dagger$ and the theorem follows.

□

where $L^\dagger$ represents the Moore-Penrose pseudoinverse of L. At the threshold a rough lower-bound for $\lambda_2(L)$ is

$$
\lambda_2(L) = \frac{2}{N} c^* \geq \min\{\lambda_2(L_1), \lambda_2(L_2)\}. \tag{A.14}
$$

One way to see this is to observe that:

$$\frac{u_1^T L_1 u_1 + u_2^T L_2 u_2}{\|u_1 + u_2\|^2} \geq \frac{\|u_1\|^2 + \|u_2\|^2}{\|u_1 + u_2\|^2} \min\{\lambda_2(L_1), \lambda_2(L_2)\}.$$

Inequality (A.14) then follows from the parallelogram law[146]. An upper bound for $\lambda_2(L)$ is given in[140]

$$\lambda_2(L) \leq \frac{1}{2}\lambda_2(L_1 + L_2). \tag{A.15}$$

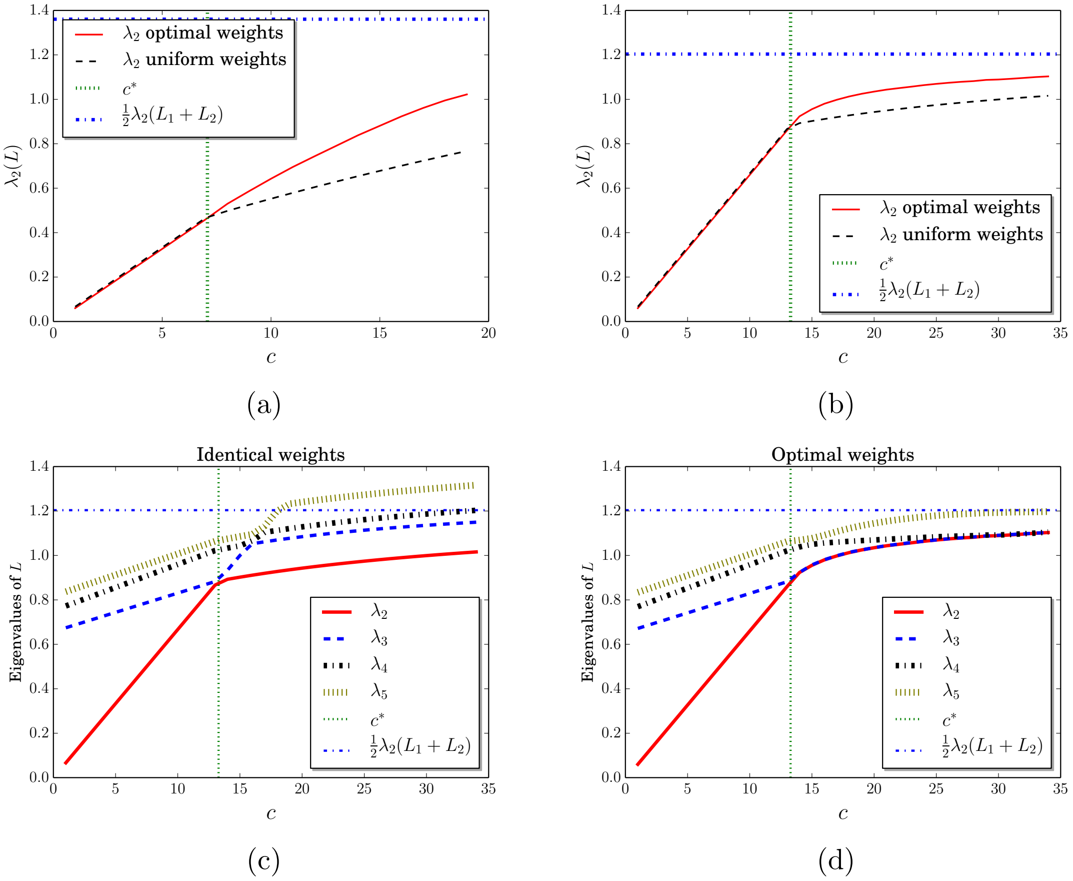
A.3 Results

In the special case of identical layers ($L_1 = L_2$) with corresponding nodes connected, the bound in (A.15) is attained with uniform weights at the threshold budget c^*[142]. This can be seen by combining (A.14) and (A.15). Therefore, in this case, uniform weights are optimal for budgets $c \leq c^*$, and increasing the budget beyond c^* cannot increase the algebraic connectivity, regardless of the weight allocation.

For general structures, it is possible to substantially improve the algebraic connectivity by increasing the budget beyond c^* using an optimal weight distribution. Figs. A.2a and A.2b compare the optimal value of $\lambda_2(L)$ to the one obtained by the uniform distribution as the budget c varies for two different network structures. In both cases, the optimal distribution gives a higher algebraic connectivity after the threshold.

In Fig. A.2c, we plot the first five eigenvalues of L (omitting the zero eigenvalue) for a multiplex with identical weights on the inter-layer links. Because $\frac{2c}{N}$ is always an eigenvalue and $\lambda_3(L) > \frac{2c}{N}$ for $c \to 0$, increasing c, $\lambda_2(L)$ and $\lambda_3(L)$ cross. For the same multiplex with optimal distribution of inter-layer weights, we plot the eigenvalues in Fig. A.2d. When increasing the budget beyond the threshold; we observe that, in this example, the second and third eigenvalues coalesce and are less than $\frac{2c}{N}$. Since (A.4) is a convex optimization problem, we know the optimal w_i's vary continously with c, and smooothly away from the finite set of budgets where eigenvalue multiplicities change.

When $c \leq c^*$, the Fiedler vector is $v = \frac{1}{\sqrt{2N}}[\mathbf{1}, -\mathbf{1}]$ and the Fiedler cut distinguishes the



(a)

(b)

(c)

(d)

Figure A.2: (a) and (b) Plots of $\lambda_2(L)$ with different amount of available budget. The solid (red) line is for the optimal weights and the dashed (black) line is for uniform weights. The threshold budget and upper-bound is shown with vertical (green) dotted and horizontal (blue) dot-dashed lines respectively. The upper-bound is from Eq. (A.15) and the threshold is from Eq. (A.11). (a) A structure of two Erdös-Renyi networks each with 30 nodes and (b) a structure of two scale-free networks each with 30 nodes. (c) First five eigenvalues of Laplacian matrix of $\mathcal{G}$ considering a uniform distribution of weights for the multiplex in (b). (d) First five eigenvalues of Laplacian matrix of $\mathcal{G}$ considering an optimal distribution of weights for the multiplex in (b).

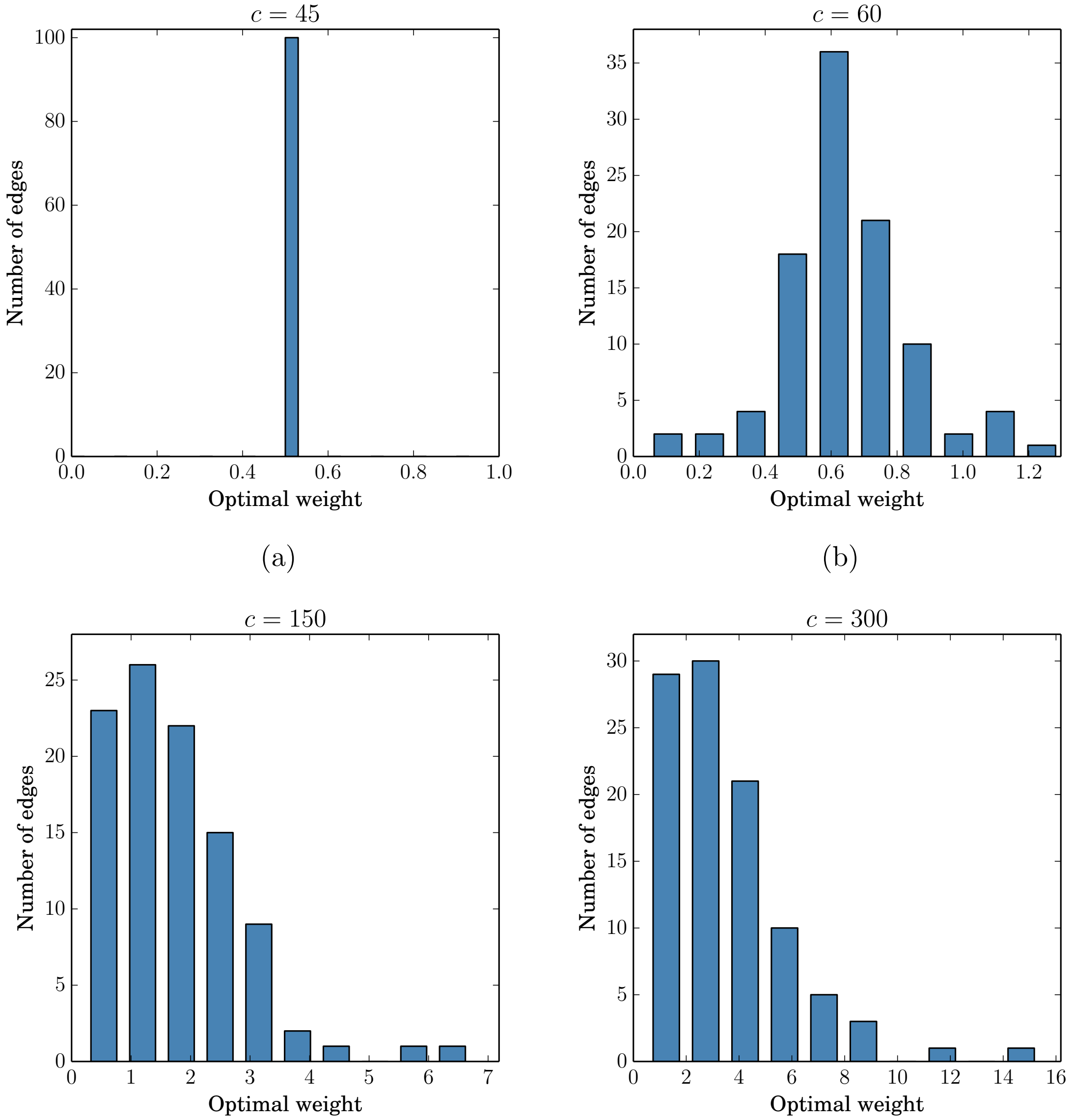
layers[142–144]. For $c > c^*$, due to the multiplicity of $\lambda_2(L)$, there is a corresponding Fiedler eigenspace. Therefore, the two layers are not as easily recognizable as before.

In Fig. A.2, we also observe that for $c > c^*$, λ_2 increases more slowly. Moreover, as Fig. A.3 shows, for a multiplex of two scale free network layers (more results in Fig. A.4 in the Appendix), we can have very non-uniform weights in this case.

These optimal weights represent the importance of each link in improving the algebraic connectivity of the whole network.

In Figure A.4, we plot the optimal weight distribution for a multiplex of two Erdös-Renyi network layers.

In summary, we have shown that before a threshold budget, the largest possible algebraic connectivity is a linear function of the budget and can only be attained by the uniform weight distribution. Since the threshold budget is always strictly positive, for low enough budgets it is not necessary to solve (A.4). On the other hand, for larger budgets, (A.4) can be solved with efficient semi-definite programming solvers to find the optimal weights. In particular, heuristic methods based solely on the information of each layer are too blunt to notice this threshold phenomenon.



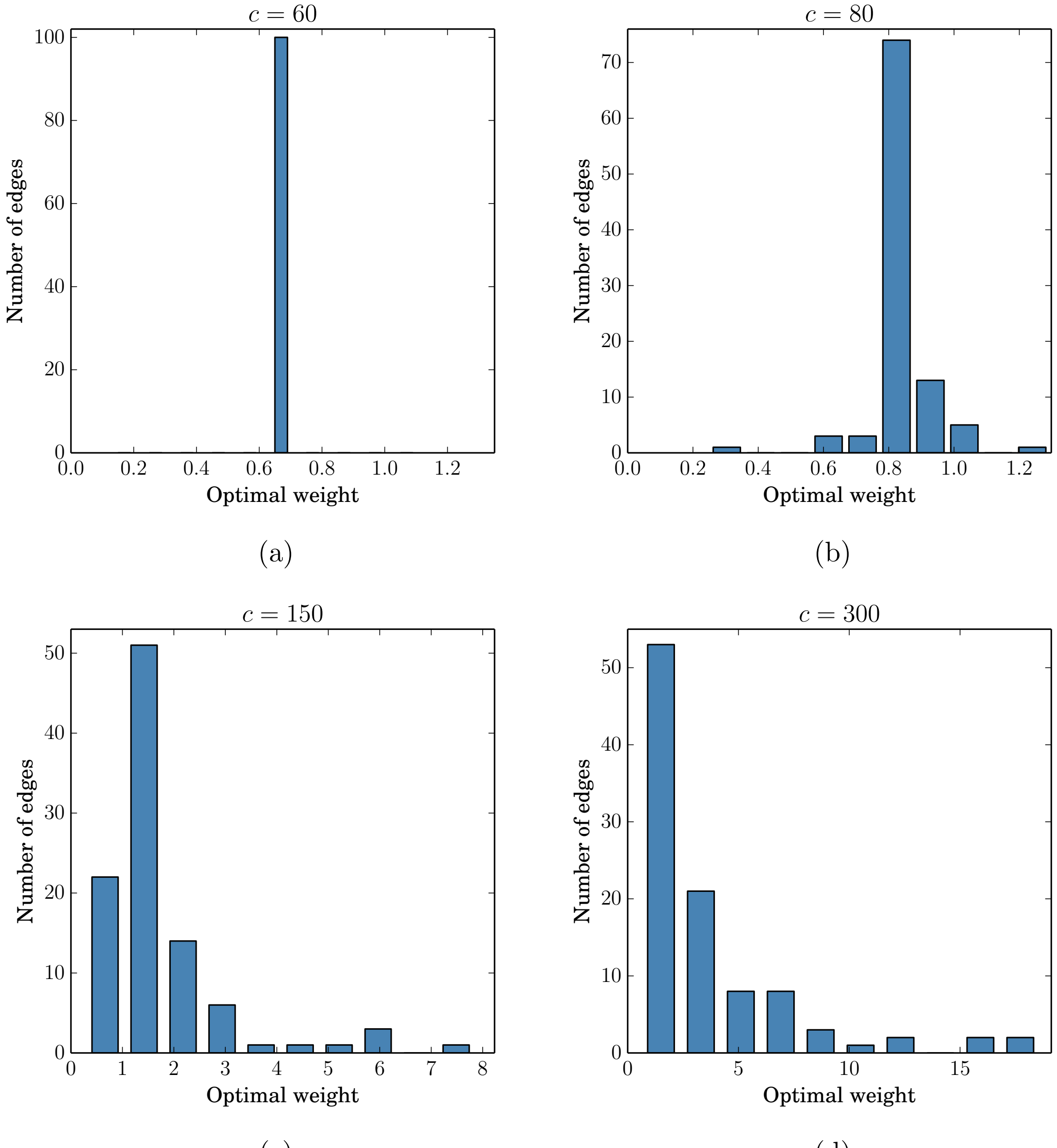
(a)

(b)

(c)

(d)

Figure A.3: Optimal weight distribution for different amount of budgets. The stucture of a multiplex with two scale free network layers, with $N = 100$ nodes and $|\mathcal{E}_1| = 196$ and $|\mathcal{E}_2| = 291$. In (a) budget is lower than threshold and uniform distribution is optimal. In this example, the threshold budget c^* is 51.4.



(a)

(b)

(c)

(d)

Figure A.4: Optimal weight distribution for different amount of budgets. The stucture of a multiplex with two scale free network layers, with $N = 100$ nodes and $|\mathcal{E}_1| = 358$ and $|\mathcal{E}_2| = 362$. In (a) budget is lower than threshold and uniform distribution is optimal. In this example, the threshold budget c^* is 64.

Appendix B

Optimal Information Dissemination Strategy to Promote Preventive Behaviors in Multilayer Epidemic Networks[2]

Launching a prevention campaign to contain the spread of infection requires substantial financial investments; therefore, a trade-off exists between suppressing the epidemic and containing costs. Information exchange among individuals can occur as physical contacts (e.g., word of mouth, gatherings), which provide inherent possibilities of disease transmission, and non-physical contacts (e.g., email, social networks), through which information can be transmitted but the infection cannot be transmitted. Contact network (CN) incorporates physical contacts, and the information dissemination network (IDN) represents non-physical contacts, thereby generating a multilayer network structure. Inherent differences between these two layers cause alerting through CN to be more effective but more expensive than IDN. The constraint for an epidemic to die out derived from a nonlinear Perron-Frobenius problem that was transformed into a semi-definite matrix inequality and served as a constraint for a convex optimization problem. This method guarantees a dying-out epidemic by choosing the best nodes for adopting preventive behaviors with minimum monetary resources. Various numerical simulations with network models and a real-world social network validate our method.

B.1 Introduction

Complications associated with modeling and analyzing epidemic spreading processes are well-studied problems. This paper focuses on mitigation of epidemic spreading, including consideration of available resources. Research in[147] and[148] showed that human behavior influences the spreading trend of an epidemic. These works introduced an extension of the "Susceptible-Infected-Susceptible" (SIS) model by adding an "Alert" state that incorporates preventive behavior. Sahneh *et al.* revealed an operating region in which the infection eventually dies out due to cautious behavior of people exposed to infected neighbors. Consequently, if an epidemic is stronger than the SIS classical threshold, long-term disease elimination is possible after a break-out period.

Sun *et al.* used an SI model to study causes of disease extinction, such as infection rate and migration[149]. In[150], Sun studied disease transmission and spatial patterns of spreading with nonlinear incidence rates. He demonstrated the positive correlation of force of infection β on these patterns.

Granell *et al.* studied interplay between disease and information in a two-layer network consisting of one physical contact network that spread the disease and a virtual overlay network that spread information to mitigate the disease[151]. They found a meta-critical point for the epidemic depending on awareness dynamics and the overlay network structure.

A majority of works concerning epidemic models have been conducted on a single graph. However, the study of disease spread in physical systems requires an elaborate interaction model based on multiple interconnected networks ([152–156]). Also[157] contains a comprehensive review on structural and dynamical organization of multilayer networks.

Sahneh *et al.* extended their analysis for multilayer networks in[19] by considering an additional directed network layer with nodes identical to the contact network (CN) but with different edges between these nodes. Information exchange was realized through these networks and each individual became aware of the state of infected neighbors at rates proportional to the number of neighbors. They proposed an optimal structure for information dissemination network (IDN) by introducing an information dissemination metric.

Preciado *et al.* controlled the spreading process by investing in alertness rates using the "Susceptible-Alert-Infected-Susceptible" (SAIS) model and considering some realistic assumptions on the cost function in order to obtain a convex optimization framework. In[158], Preciado *et al.* attempted to ensure that largest eigenvalue was smaller than the persisting threshold introduced in[148], consequently leading to rate control based on CN structure.

Motivated by[158] and using threshold concepts in[147;148], we attempted to identify alertness rates on multilayer networks in order to achieve a dying-out epidemic. However this problem is more general than[19] because each layer can have an arbitrary structure. The second threshold was obtained from a nonlinear eigenvalue problem that is a nonlinear form of the Perron-Frobenius problem. In order to obtain optimal rates, we coupled this nonlinear Perron-Frobenius problem (NPF) with a convex optimization problem, creating a general method that can be applied to solve a variety of optimization problems combined with NPF problems in various disciplines. Optimal rates were obtained for a specific effective infection rate, so epidemics with identical or weaker effective infection rates will certainly die-out with a safety margin. In addition, by monitoring the status of a small subgroup and characterizing epidemic properties and behavioral response, we obtained a cost effective strategy to mitigate long run spreading for the entire population.

The remainder of the paper is organized as follows. First, we introduce our notation and modeling method and we analyze characteristics of the multilayer model. In Section B.4, we introduce problem statements, and in Section B.5 we demonstrate how to approach this problem, proving necessary properties and introducing the coupled NPF problem with the convex problem. In Section B.6, we solve several examples of standard networks and a real-world network and discuss results.

B.2 Multilayer Network Structure

We used a multilayer network structure to represent multiple types of interconnection among individuals in the population. A multilayer network consists of L layers of graphs that have identical nodes but their edges can be different and independently formed. In this work, we considered a two-layer network. Although a disease can propagate among individuals through the physical contact network (CN), information can spread among the same individuals through an on-line information dissemination network (IDN).

Since physical interactions can be considered as undirected edges and we omit individuals who do not interact with the population, therefore, these assumptions lead to an undirected and connected graph for CN. Some people may not have a social network account or a person may follow a celebrity on Twitter but that celebrity does not reciprocate; therefore, IDN can be directed and not connected.

$A = [a_{ij}] \in \mathbb{R}^{N \times N}$ denotes the adjacency matrix of CN, where $a_{ij} = 1$ if and only if $(i, j) \in \mathcal{E}$; otherwise $a_{ij} = 0$. Similarly, we defined the adjacency matrix of IDN as $B = [b_{ij}]_{N \times N}$. The largest eigenvalue of the adjacency matrix A, known as the spectral radius of A, is denoted by $\lambda_1(A)$; elements of the corresponding eigenvector v_1 are real and non-negative. Spectral centrality of nodes in a graph is determined by the rank of corresponding elements of v_1.

B.3 Model Development

In this paper, results are based on the SAIS model developed in[148]. Each node is allowed to be in one of three states: *'susceptible'*, *'infected'*, or *'alert'* and a node maintained the same state in all layers. A susceptible node becomes infected with a given infection rate through infected neighbors in CN and becomes alert through infected neighbors in different layers with corresponding rates. An alert node becomes infected with a rate less than the initial infection rate. An infected node is recovered at a given removing/recovery rate. For each agent $i \in \{1, ..., N\}$, let the random variable $x_i(t) = e_1$, if the agent i is susceptible at time t, $x_i(t) = e_2$ if alert, and $x_i(t) = e_3$ if infected, where $e_1 = [1,0,0]^T$, $e_2 = [0,1,0]^T$, and $e_3 = [0,0,1]^T$ are standard unit vectors of $\mathbb{R}^3$. Throughout this paper, the infection rate for an alert individual is assumed to be a reduced version of β, i.e., $r\beta$ with $r \leq 1$.

In the following equations, $\Pr[\cdot]$ denotes probability, $X(t) \triangleq \{x_i(t), i = 1, ..., N\}$ is the joint state of the network, $\Delta t > 0$ is a time step, and the indicator function $1_{\{\mathcal{X}\}}$ is 1 if $\mathcal{X}$ is true and 0 otherwise. A function $f(\Delta t)$ is said to be $o(\Delta t)$ if $\lim_{\Delta t \to 0} \frac{f(\Delta t)}{\Delta t} = 0$. For node i, $Y_i\left(t\right)$ is the number of neigbors in CN who are infected at time t and Z_i is the number of neigbors in IDN who are infected at time t:

$$Y_i\left(t\right) \triangleq \sum_{j=1}^{N} a_{ij} 1_{\{x_j(t)=e_3\}},$$

$$Z_i(t) \triangleq \sum_{j=1}^{N} b_{ij} 1_{\{x_j(t)=e_3\}}.$$

There are four stochastic transitions in the SAIS model:

1. A susceptible agent becomes infected with infection rate β times the number of infected neighbors:

$$\Pr\left[x_i\left(t+\Delta t\right) = e_3 | x_i\left(t\right) = e_1, X\left(t\right)\right] = \beta Y_i\left(t\right)\Delta t + o\left(\Delta t\right), \tag{B.1}$$

   for $i \in \{1, ..., N\}$.

2. An infected agent recovers to the susceptible state with curing rate δ:

$$\Pr\left[x_i\left(t+\Delta t\right) = e_1 | x_i\left(t\right) = e_3, X\left(t\right)\right] = \delta\Delta t + o\left(\Delta t\right). \tag{B.2}$$

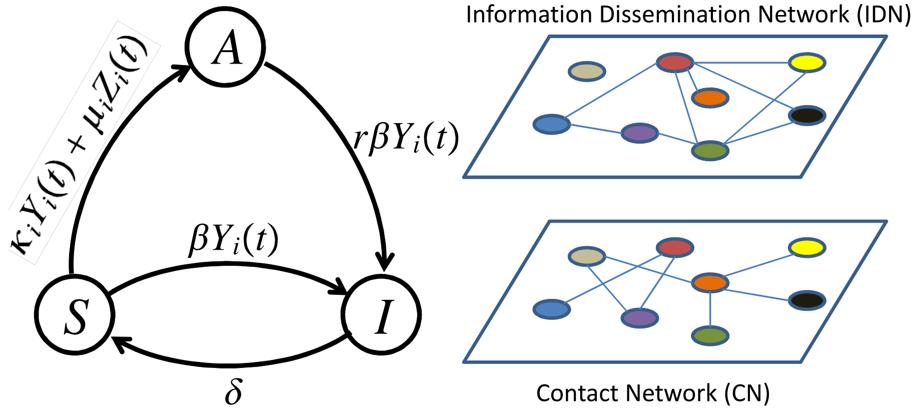
3. A susceptible agent may become alert if surrounded by infected individuals in both CN and IDN. Specifically, a susceptible node becomes alert with alerting rate $\kappa \in \mathbb{R}^+$ times the number of infected neighbors in CN and with alerting rate $\mu \in \mathbb{R}^+$ times the number of infected neighbors in IDN:

$$\Pr\left[x_i\left(t+\Delta t\right)=e_2|x_i\left(t\right)=e_1,X\left(t\right)\right]=\left(\kappa_i Y_i\left(t\right)+\mu_i Z_i\left(t\right)\right)\Delta t+o\left(\Delta t\right), \qquad \text{(B.3)}$$

4. An alert agent can become infected but with a weaker infection rate $0<r\beta<\beta$:

$$\Pr\left[x_i\left(t+\Delta t\right)=e_3|x_i\left(t\right)=e_2,X\left(t\right)\right]=r\beta Y_i\left(t\right)\Delta t+o\left(\Delta t\right). \qquad \text{(B.4)}$$

Stochastic compartmental transitions of a node are depicted in Figure B.1-a. An Illustrative schematic of CN and IDN is shown in Figure B.1-b.



A
$\kappa_i Y_i(t)+\mu_i Z_i(t)$
$r\beta Y_i(t)$
$\beta Y_i(t)$
S
I
δ
Information Dissemination Network (IDN)
Contact Network (CN)

Figure B.1: From left to right, (a) Compartmental transition graph according to the SAIS model with information dissemination. Y_i and Z_i are the number of infected neighbors of agent i in contact network and information dissemination network, respectively[19]; (b) Multilayer contact topology.

Let p_i and q_i denote the probabilities that agent i is infected and alert, respectively. The SAIS model with the information dissemination layer is obtained with some modification from[19]:

$$\dot{p}_i=\beta\left(1-p_i-q_i\right)\sum_{j=1}^{N}a_{ij}p_j+r\beta q_i\sum_{j=1}^{N}a_{ij}p_j-\delta p_i; \qquad \text{(B.5)}$$

$$\dot{q}_i=\left(1-p_i-q_i\right)\left\{\kappa_i\sum_{j=1}^{N}a_{ij}p_j+\mu_i\sum_{j=1}^{N}b_{ij}p_j\right\}-r\beta q_i\sum_{j=1}^{N}a_{ij}p_j. \qquad \text{(B.6)}$$

Equations (B.5) and (B.6) are derived by a mean field Type approximation.

B.3.1 Analysis of SAIS Model

SAIS with No Alertness (SIS)

When no alertness transmission is present through CN or IDN, $\kappa_i = 0$ and $\mu_i = 0$, the model reduced to the original SIS model, as discussed in[147]. Therefore, the system exhibits a threshold for the effective infection rate $\tau \triangleq \frac{\beta}{\delta}$, under which the infection dies out exponentially. This threshold has been proven to be the inverse of the largest eigenvalue of CN adjacency matrix in $\tau_{c_1} \triangleq \frac{1}{\lambda_1(A)}$.

SAIS with Alertness Dissemination

Theorem B.3.1. *In the SAIS model (B.5-B.6), initial infections will die out exponentially if the effective infection rate τ is less than $\tau_{c_1} = \lambda_1^{-1}$ and a second threshold, τ_{c_2}, exists such that if $\tau_{c_1} < \tau < \tau_{c_2}$, then the infection dies out asymptotically after an initial spread. In addition, the second threshold $\tau_{c_2}(\kappa_i, \mu_i)$ is a monotonically increasing function of κ_i and μ_i.*

Proof. [19] contains the proof. □

The first threshold depends only on topology of the CN layer, but the second threshold depends on behavioral properties and topology of both layers.

After the second threshold, i.e., $\tau > \tau_{c_2}$, steady-state values of infection probabilities are positive and τ_{c_2} can be determined by studying the steady-state solution. According to (B.5) and (B.6), at the steady-state,

$$(1 - p_i^*) \left\{ \kappa_i \sum_{j=1}^{N} a_{ij} p_j^* + \mu_i \sum_{j=1}^{N} b_{ij} p_j^* \right\} - q_i^* \left\{ \kappa \sum_{j=1}^{N} a_{ij} p_j^* + k \sum_{j=1}^{N} b_{ij} p_j^* \right\} - r\beta q_i^* \sum_{j=1}^{N} a_{ij} p_j^* = 0; \quad \text{(B.7)}$$

$$q_i^* = (1-p_i^*)\frac{\bar{\kappa}_i \sum_{j=1}^N a_{ij}p_j^* + \bar{\mu}_i \sum_{j=1}^N b_{ij}p_j^*}{(1+\bar{\kappa}_i)\sum_{j=1}^N a_{ij}p_j^* + \bar{\mu}_i \sum_{j=1}^N b_{ij}p_j^*}, \tag{B.8}$$

where p_i^* and q_i^* are steady-state probabilities and $\bar{\kappa}_i \triangleq \frac{\kappa_i}{r\beta}$ and $\bar{\mu} \triangleq \frac{\mu}{r\beta}$ are normalized alertness rates. Combining (B.7) and (B.8), the steady-state equation becomes,

$$\begin{aligned} &\tau(1-p_i^*)\sum_{j=1}^N a_{ij}p_j^* \\ &\qquad - \tau(1-r)(1-p_i^*)\frac{\bar{\kappa}_i \sum_{j=1}^N a_{ij}p_j^* + \bar{\mu}_i \sum_{j=1}^N b_{ij}p_j^*}{(1+\bar{\kappa}_i)\sum_{j=1}^N a_{ij}p_j^* + \bar{\mu}_i \sum_{j=1}^N b_{ij}p_j^*}\sum_{j=1}^N a_{ij}p_j^* \\ &\qquad\qquad = p_i^*. \end{aligned} \tag{B.9}$$

Theorem B.3.2. *The second threshold is the nontrivial solution of the following nonlinear eigenvalue problem:*

$$\tau_{c_2} diag\left(\frac{(1+r\bar{\kappa}_i)\sum_{j=1}^N a_{ij}w_j + r\bar{\mu}_i \sum_{j=1}^N b_{ij}w_j}{(1+\bar{\kappa}_i)\sum_{j=1}^N a_{ij}w_j + \bar{\mu}_i \sum_{j=1}^N b_{ij}w_j}\right) A_{\mathcal{G}}\mathbf{w} = \mathbf{w}, \tag{B.10}$$

where $\mathbf{w} = [w_1, ..., w_N]^T$, *with* $w_i > 0 \quad \forall i = 1, ..., N$.

Proof. Define $\tilde{\tau} \triangleq \tau - \tau_{c_2}$. Close to the second threshold, i.e., as $\tilde{\tau} \to 0^+$, we have $p_i^* = \tilde{\tau}\frac{\partial p_i^*}{\partial \tau}|_{\tau=\tau_{c_2}} + o(\tilde{\tau})$. Letting $\tau \to \tau_{c_2}^+$ in (B.9),

$$\begin{aligned} &\tau_{c_2}\sum_{j=1}^N a_{ij}\frac{\partial p_j^*}{\partial \tau}|_{\tau=\tau_{c_2}} \\ &\qquad - \tau_{c_2}(1-r)\frac{\bar{\kappa}_i \sum_{j=1}^N a_{ij}\frac{\partial p_j^*}{\partial \tau}|_{\tau=\tau_{c_2}} + \bar{\mu}_i \sum_{j=1}^N b_{ij}\frac{\partial p_j^*}{\partial \tau}|_{\tau=\tau_{c_2}}}{(1+\bar{\kappa}_i)\sum_{j=1}^N a_{ij}\frac{\partial p_j^*}{\partial \tau}|_{\tau=\tau_{c_2}} + \bar{\mu}_i \sum_{j=1}^N b_{ij}\frac{\partial p_j^*}{\partial \tau}|_{\tau=\tau_{c_2}}} \\ &\qquad\qquad \sum_{j=1}^N a_{ij}\frac{\partial p_j^*}{\partial \tau}|_{\tau=\tau_{c_2}} = \frac{\partial p_i^*}{\partial \tau}|_{\tau=\tau_{c_2}}. \end{aligned} \tag{B.11}$$

Because τ_{c_2} is the second threshold, $\frac{\partial p_i^*}{\partial \tau}|_{\tau=\tau_{c_2}}$ must be positive for every $i \in \{1, ..., N\}$.

Therefore, τ_{c_2} is such that the set of algebraic equations (B.10) has positive solutions. By substituting $w_j = \frac{\partial p_j^*}{\partial \tau}|_{\tau=\tau_{c_2}}$ in (B.11), the nonlinear eigenvalue problem in (B.10) can be obtained. □

B.4 Problem Statement

Given network layer adjacency matrices A and B and disease properties β, δ, and r, the following optimization problem is considered:

$$\begin{aligned} \underset{\bar{\kappa}_i,\bar{\mu}_i}{\text{minimize}} \quad & \sum_{i=1}^{N} f_i\left(\bar{\kappa}_i, \bar{\mu}_i\right) \\ \text{subject to} \quad & \tau_{c_2}\left(\bar{\kappa}_i, \bar{\mu}_i\right) \geq \tau, \\ & \bar{\mu}_{min} \leq \bar{\mu}_i \leq \bar{\mu}_{max}, \\ & \bar{\kappa}_{min} \leq \bar{\kappa}_i \leq \bar{\kappa}_{max}. \end{aligned} \tag{B.12}$$

where f_i is a linear fractional cost function to promote alertness in the population[158]. Minimization of this objective function while constraining the system to die out asymptotically, requires a trade-off.

Asymptotic Stability Constraint

According to Theorem B.3.1, in order to have an asymptotically dying-out infection, the effective infection rate should be less than the second threshold, corresponding to the first constraint in (B.12).

The nonlinear eigenvalue problem in (B.10) can be written as follows:

$$diag\left(h_i\left(\boldsymbol{\xi}_i, \mathbf{w}\right)\right) A\mathbf{w} = \lambda \mathbf{w}, \tag{B.13}$$

where $diag\left(h_i\left(\boldsymbol{\xi}_i, \mathbf{w}\right)\right)$ is a nonnegative diagonal matrix with unknown parameters $\boldsymbol{\xi}_i = \left[\bar{\kappa}_i, \bar{\mu}_i\right], \quad \forall i = 1, ..., N$ and λ corresponds to the inverse of the second threshold τ_{c_2}. Therefore the eigenvalue problem in (B.13) is a NPF problem[159].

According to NPF theory, the largest eigenvalue is positive and real and the corresponding normalized eigenvector $\mathbf{w}$ is unique and positive[159].

B.5 Solution Methodology

Given $\boldsymbol{\xi}_i$ using the power iteration algorithm, the largest eigenvalue and corresponding eigenvector of (B.13) can be found. Starting with an initial guess for $\mathbf{w^{(0)}}$, the following iteration is performed:

$$diag\left(h_i\left(\boldsymbol{\xi}_i, \mathbf{w}^{(l)}\right)\right) A\mathbf{w}^{(l)} = \lambda^{(l+1)}\mathbf{w}^{(l+1)}, \tag{B.14}$$

where $\lambda^{\mathbf{(k+1)}}$ and $\mathbf{w^{(k+1)}}$ are the approximated value of the largest eigenvalue and corresponding eigenvector in the k'th step. They are obtained from the following relations:

$$\lambda^{(l+1)} = \parallel diag\left(h_i\left(\boldsymbol{\xi}_i, \mathbf{w}^{(l)}\right)\right) A\mathbf{w}^{(l)} \parallel; \tag{B.15}$$

$$\mathbf{w}^{(l+1)} = \frac{diag\left(h_i\left(\boldsymbol{\xi}_i, \mathbf{w}^{(l)}\right)\right) A\mathbf{w}^{(l)}}{\parallel diag\left(h_i\left(\boldsymbol{\xi}_i, \mathbf{w}^{(l)}\right)\right) A\mathbf{w}^{(l)} \parallel}. \tag{B.16}$$

This algorithm has guaranteed convergence to the largest eigenvalue and corresponding eigenvector of (B.13) with a chosen tolerance ε. Pseudocode for this algorithm is given in Algorithm 1.

Algorithm 6 Power iteration

```
Require: guess← $\mathbf{w^{(0)}}$
Ensure: $\mathbf{w} = \mathbf{w}^{(l+1)}$
for $l$ do
    $\mathbf{w}^{(l)}$ ←guess
    $\mathbf{w}^{(l+1)} = \frac{diag\left(h_i\left(\boldsymbol{\xi}_i,\mathbf{w}^{(l)}\right)\right)A\mathbf{w}^{(l)}}{\|diag\left(h_i\left(\boldsymbol{\xi}_i,\mathbf{w}^{(l)}\right)\right)A\mathbf{w}^{(l)}\|}$
    if $\mid \mathbf{w}^{(l+1)} - \mathbf{w}^{(l)} \mid \leq \varepsilon$ then
        stop
    end if
    guess← $\mathbf{w}^{(l+1)}$
end for
```

Proposition B.5.1. *Optimal parameters in (B.10) can be found by alternating between the NPF problem and the optimization problem.*

Proof. Starting with a guess for $\boldsymbol{\xi}_i^{(0)}$,

$$diag\left(h_i\left(\boldsymbol{\xi}_i^{(0)}, \mathbf{w}\right)\right) A\mathbf{w} = \lambda \mathbf{w}. \tag{B.17}$$

Using the derived eigenvector $\mathbf{w^{(0)}}$ from the power method, we approximate $diag\left(h_i\left(\boldsymbol{\xi}_i, \mathbf{w}\right)\right)$ as $diag\left(h_i\left(\boldsymbol{\xi}_i, \mathbf{w}^{(0)}\right)\right)$ and then solve the optimization problem in (B.12), and find new approximation for parameters $\boldsymbol{\xi}_i^{(1)}$ as the new guess. Because of existing constraints, new obtained parameters ensure initial NPF problem properties. Using the updated guess, we alternate between the NPF problem and the optimization problem until these guesses converges with a tolerance ϵ, i.e., $\mid \boldsymbol{\xi}_i^{(k)} - \boldsymbol{\xi}_i^{(k-1)} \mid \leq \epsilon$. □

At each step with approximated $diag\left(h_i\left(\boldsymbol{\xi}_i, \mathbf{w}^{(k)}\right)\right)$, the NPF problem in B.10 becomes a linear Perron-Frobenius problem. Therefore, the first constraint in (B.12) transforms to a semidefinite inequality with the following lemma.

Lemma B.5.2. *If D is a diagonal matrix with positive diagonal entries, A is a symmetric matrix, τ_c and τ are scalars, and the following eigenvalue problem exists:*

$$\tau_c D A w = w, \tag{B.18}$$

for $\tau \leq \tau_c$:

$$A - (\tau D)^{-1} \preceq 0. \tag{B.19}$$

Proof. First, we show that eigenvalues of $\tau_c DA$ are real with the following variable change:

$$w = D^{\frac{1}{2}} x. \tag{B.20}$$

Rewriting (B.18),

$$\tau_c DAD^{\frac{1}{2}}x = D^{1/2}x, \tag{B.21}$$

and multiplying both sides by $D^{-\frac{1}{2}}$ produces

$$\tau_c D^{\frac{1}{2}}AD^{\frac{1}{2}}x = x, \tag{B.22}$$

which shows that DA and $D^{\frac{1}{2}}AD^{\frac{1}{2}}$ share similar eigen-properties. Then, since $D^{\frac{1}{2}}AD^{\frac{1}{2}}$ is symmetric, it has real eigenvalues; therefore, DA also has real eigenvalues.

From (B.18):

$$\lambda_1\left(\tau_c DA\right) = 1. \tag{B.23}$$

If $\tau \leq \tau_c$

$$\tau\lambda_1\left(DA\right) - 1 \leq 0,$$

which can be rewritten as

$$\lambda_1\left(\tau DA - I\right) \leq 0. \tag{B.24}$$

Equations (B.24) and (B.22) show that

$$\left(\tau D^{\frac{1}{2}}AD^{\frac{1}{2}} - I\right) \preceq 0, \tag{B.25}$$

or

$$A - \left(\tau D\right)^{-1} \preceq 0. \tag{B.26}$$

□

Convex Formulation

According to Lemma B.5.2, the dying-out constraint, $\tau_{c_2}\left(\bar{\kappa}_i, \bar{\mu}_i\right) \geq \tau = \frac{\beta}{\delta}$ can be written as,

$$A - \frac{\delta}{\beta} diag\left(\frac{\left(1+\bar{\kappa}_i\right)\phi_i^A + \bar{\mu}_i \phi_i^B}{\left(1+r\bar{\kappa}_i\right)\phi_i^A + r\bar{\mu}_i \phi_i^B}\right) \preceq 0, \tag{B.27}$$

where ϕ_i^A and ϕ_i^B represent $\sum_{j=1}^{N} a_{ij} w_j$ and $\sum_{j=1}^{N} b_{ij} w_j$, respectively. For a linear fractional cost function,

$$\sum_{i=1}^{N} \frac{c_i \bar{\kappa}_i + t_i \bar{\mu}_i}{r\bar{\kappa}_i \phi_i^A + r\bar{\mu}_i \phi_i^B + \phi_i^A}, \tag{B.28}$$

the problem in (B.12) is a quasiconvex optimization problem[62].

Because all equations are homogeneous, we choose a scale z_i such that for $i \in 1, \cdots, N$, $z_i\left(r\bar{\kappa}_i \phi_i^A + r\bar{\mu}_i \phi_i^B + \phi_i^A\right) = 1$. Substituting[1] $u_i = z_i \bar{\kappa}_i$ and $v_i = z_i \bar{\mu}_i$ produced the following semi-definite optimization problem (SDP) equivalent to (B.12):

$$\begin{aligned}
\underset{u_i, v_i, z_i}{\text{minimize}} \quad & \sum_{i=1}^{N} \left(c_i u_i + t_i v_i\right) \\
\text{subject to} \quad & A - \frac{\delta}{\beta} diag\left(u_i \phi_i^A + v_i \phi_i^B + z_i \phi_i^A\right) \preceq 0, \\
& rU\Phi^A + rV\Phi^B + Z\Phi^A = I, \\
& \bar{\mu}_{min} z_i \leq u_i \leq \bar{\mu}_{max} z_i, \\
& \bar{\kappa}_{min} z_i \leq v_i \leq \bar{\kappa}_{max} z_i.
\end{aligned} \tag{B.29}$$

where U, V, Z, Φ^A, and Φ^B are diagonal matrices with u_i, v_i, z_i, ϕ_i^A and ϕ_i^B as their entries, respectively. Using classic solvers such as interior point-based methods, the SDP in (B.29) can be solved in a fast and robust fashion for networks up to 1000 nodes. In this work, we use CVX[161]. Subgradient methods or smoothing and accelerated algorithms can be used to efficiently solve (B.29) in very large networks. These methods are well-studied and powerful commercial solvers are developed for applying them.

From Proposition B.5.1, Φ^A and Φ^B update with each iteration and carry new structural

[1] This transformation is similar to Charnes-Cooper transformation[160].

properties; therefore, a new optimization problem should be solved each time causing $\bar{\kappa}_i$, $\bar{\mu}_i$, and w_i to converge to the desired solution. Pseudocode is given in Algorithm 7.

Algorithm 7 Power iteration

```
Require: guess← $\boldsymbol{\xi}_i^{(0)}$
Ensure: $\boldsymbol{\xi} = \boldsymbol{\xi}_i^{(k)}$
for $k$ do
    $\boldsymbol{\Phi}^{\mathbf{A}^{(k)}}, \boldsymbol{\Phi}^{\mathbf{B}^{(k)}}$ ←Power method
    Convex Problem← $\boldsymbol{\Phi}^{\mathbf{A}^{(k)}}, \boldsymbol{\Phi}^{\mathbf{B}^{(k)}}$
    $\boldsymbol{\xi}_i^{(k)}$ ←Convex Problem
    if $\mid \boldsymbol{\xi}_i^{(k)} - \boldsymbol{\xi}_i^{(k-1)} \mid \leq \epsilon$ then
        stop
    end if
    guess← $\boldsymbol{\xi}_i^{(k)}$
end for
```

B.6 Numerical Simulations

We considered an infectious disease with an effective infection rate $\frac{\beta}{\delta} = \frac{3}{\lambda_1(A)}$, an unstable situation in SIS, and a reduction in infection rate $r = \frac{1}{3}$ due to alertness. Alertness rates vary between an upper limit and a lower limit, based on response capacity of the population. Due to inherent differences between μ_i and κ_i, a higher awareness may be reached through CN, but these rates are more expensive than rates in IDN. In the following simulations, we assume that $\mu_{max} = 5$, $\kappa_{max} = 10$, and $\mu_{min} = \kappa_{min} = 0$, and cost function weights are $c = 1.5$ and $t = 1$.

For the following multilayer structures, if no information dissemination is available the second threshold does not exist and dying-out epidemic occurs if $\frac{\beta}{\delta} < \frac{1}{\lambda_1(A)}$. Considering information dissemination through CN and without IDN, if we assign the highest amount of alertness rate for all individuals, i.e., $\kappa_i = \kappa_{max}$, then $\tau_{c_2}\left(\kappa_i = \kappa_{max}, \mu_i = 0\right) = \frac{1+\kappa_{max}}{1+r\kappa_{max}} \frac{1}{\lambda_1(A)} = 2.51 \frac{1}{\lambda_1(A)}$ which cannot suppress the epidemic even though it is very expensive. However, use of IDN and proposed optimal rates helps achieve a cost-efficient suppression of the epidemic.

In the following simulations, we selected a preferential attachment network[6] for IDN. Four networks were selected for CN: a regular random network, a geometric random network[5], a preferential attachment network, and a real-world social (face-to-face) network.

Example 1: CN is a regular random graph

In this example, the CN layer is a regular random graph. Nodes in a regular graph has the same number of neighbors. Obtained optimal alertness rates are depicted in Figure B.2.

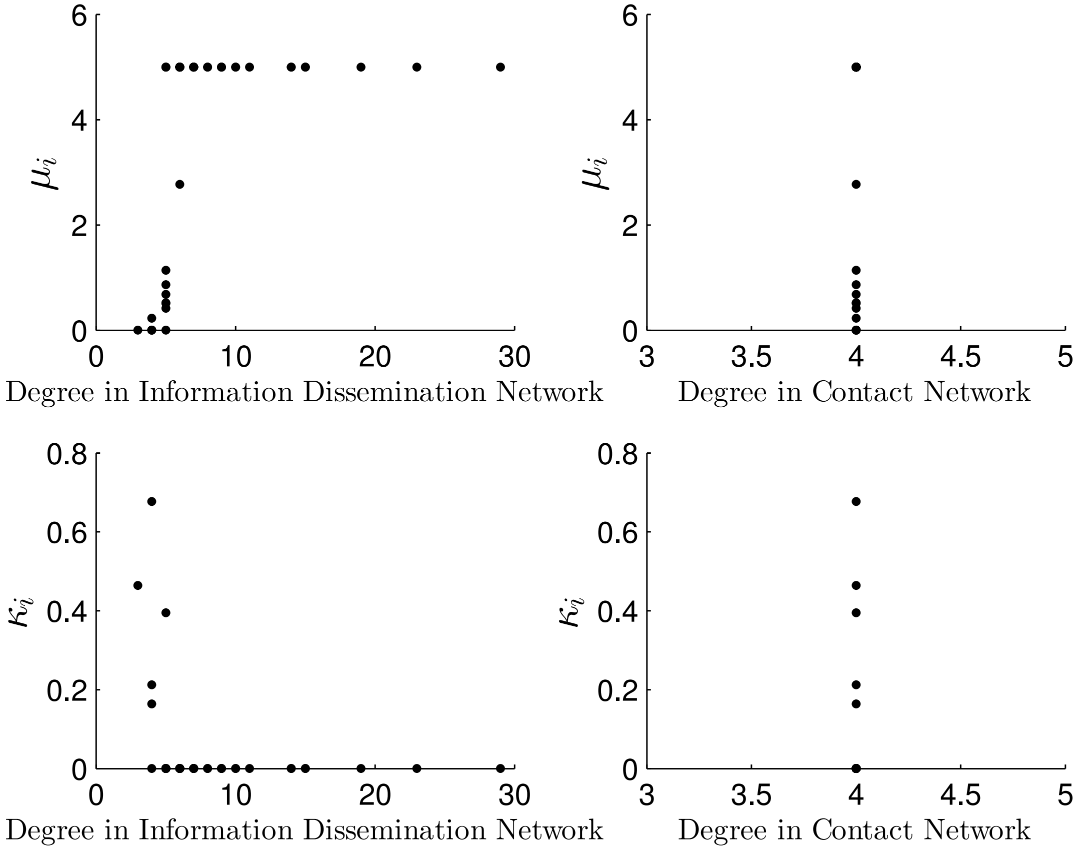


Figure B.2: Optimal alertness rates μ_i and κ_i for $i = 1, \cdots, 50$, with respect to degree of nodes in both layers. Each layer has 50 nodes. The information dissemination network is a preferential attachment network with minimum node degree 5, and the contact network has a random regular structure with node degree 4. *Note:* There are nodes with the same node degree and same optimum rates which caused overlapping in the figure.

Because all individuals have identical number of neighbors in CN, the only difference between them is their degree in IDN. For individuals with high degree, μ_i influence is more effective compared to lower degree nodes. Since promotion of μ_i is less expensive than κ_i, maximum investment of μ_i is optimum after a certain degree (nodes with $deg_i \geq 6$ in this example) in IDN. For nodes with lower degree in IDN, more reliable sources of information are neighbors in CN, similar to occasions when people are not active in online social networks and must be contacted through their neighbors in CN. In this example, since promoting alertness in CN is more expensive, although $\kappa_{max} > \mu_{max}$, the optimization problem does not allow any node to have $\kappa_i = \kappa_{max}$ while $\mu_i = \mu_{max}$.

Example 2: CN is a random geometric graph

In this example, CN is a random connected geometric graph in a two-dimensional coordinate system. Optimal alerting rates versus degree in both layers are shown in Figure B.3. Because high degree nodes in CN mean increased exposure to the infection, a high emphasis must be assigned to them. Furthermore, low degree in CN means decreased infecting opportunities and, because of limited monetary resources, the proposed method allocates all available resources to higher degree nodes. Unlike Example 1, some nodes are assigned with the maximum amount of κ_i. Nodes with $\mu_i = \mu_{max}$ and $\kappa_i = \kappa_{max}$ (saturated nodes) are hubs in IDN and high degree nodes in CN.

Example 3: CN is a random preferential attachment network

In this example, both layers are preferential attachment networks with different preferential attachment probabilities and 80 nodes. Optimal alerting rates as a function of node degree in both layers are shown in Figure B.4. Similar to previous examples investment on very low degree individuals in the presence of financial restrictions is not wise. Since high degree nodes in CN are more exposed to the infection, they must be encouraged to be alert through IDN or CN neighbors. Results identical to example 2 can be observed. Emphasis on high degree individuals is notable (hubs).

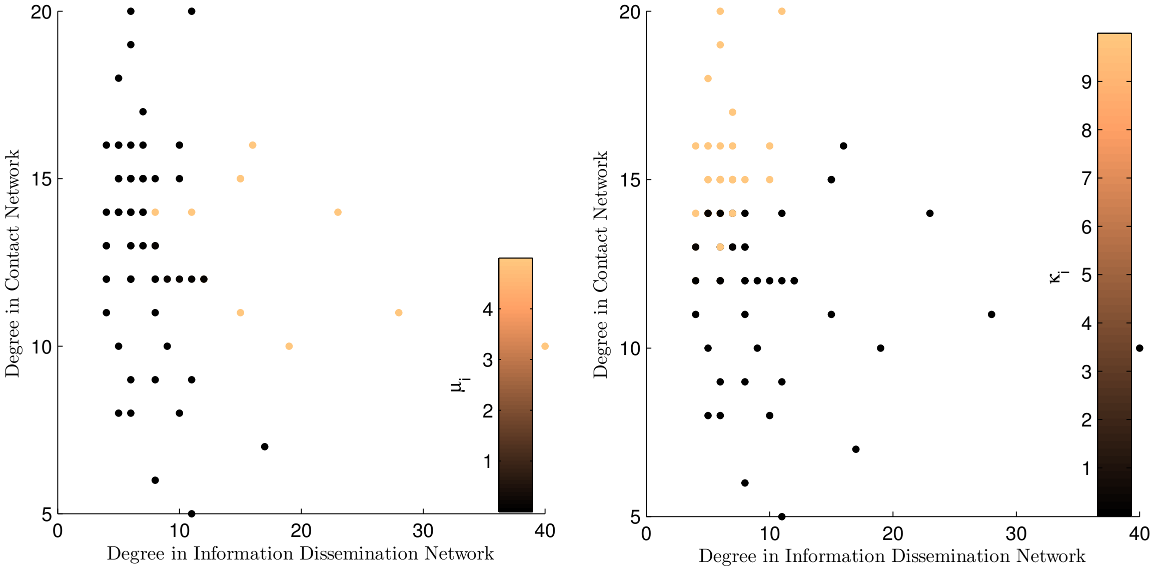


Figure B.3: Alertness rates μ_i and κ_i for $i = 1, \cdots, 80$, with respect to degree of each nodes in both layers. Each layer has 80 nodes. The IDN is a preferential attachment network with minimum node degree 5, and the contact network has a random geometric structure. *Note:* Colors represent the optimum rates according to the colorbar. Nodes with identical node degree caused overlapping in the figure.

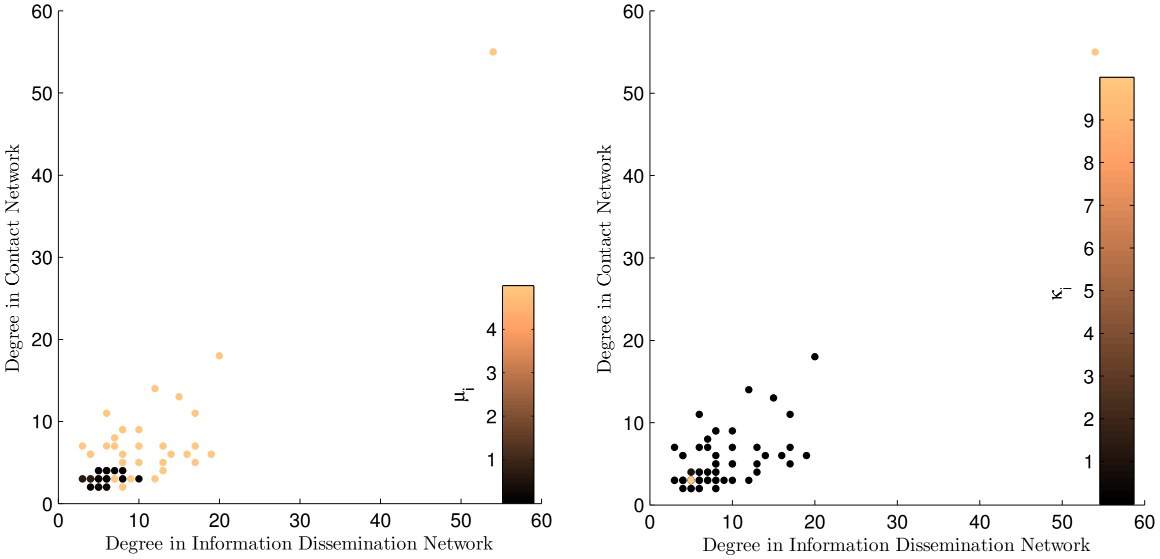


Figure B.4: Alertness rates μ_i and κ_i for $i = 1, \cdots, 80$, with respect to degree of each nodes in both layers. Each layer has 80 nodes. The IDN is a preferential attachment network with minimum node degree 5, and the contact network has a preferential network with minimum node degree 3. *Note:* Colors represent the optimum rates according to the colorbar. Nodes with identical node degree caused overlapping in the figure.

Example 4: CN is a social (face-to-face) network

In this example, the CN layer is a portion of the social contact network based on survey of a community in Chanute, Kansas, United States[20]. In an effort to consider important connections in the network, we remove links with weights less than 0.2. Based on weight distribution in[20], we consider the remining connected network as an unweighted CN with 102 nodes (Figure B.5).

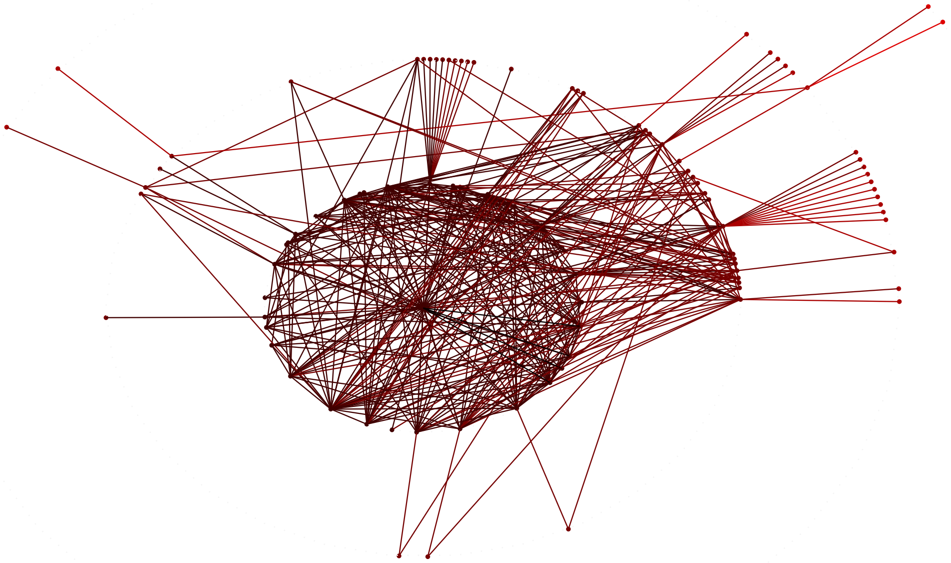


Figure B.5: A portion of the social (face-to-face) network built based on a survey of a community in Chanute, Kansas, United States[20]. Network size is 102, maximum node degree is 36, and minimum node degree is 1

Similar to previous examples a preferential attachment network as IDN with the same size as CN is considered. Optimal alerting rates as a function of node degree in both layers are shown in Figure B.6. Results identical to the previous examples are observed. In addition, for nodes with high degree in CN or more exposed to the infection, optimal alertness investments are either through CN or IDN. An extreme case, such as a hub, that must be considered from both networks was not observed in this example.

The threshold phenomena predicted in[158] is observed in these examples too. Because of effects from IDN, this threshold is not abrupt. In order to determine optimal alertness rates, a transition zone exists with a trade-off between topological characteristics of both layers.

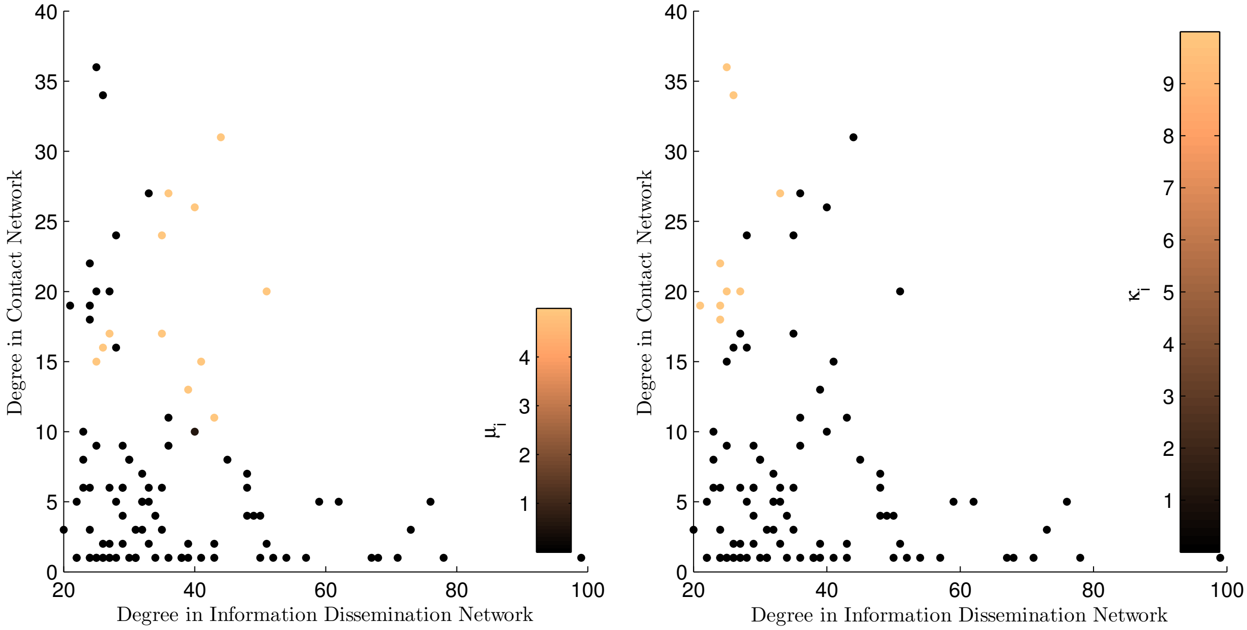


Figure B.6: Alertness rates μ_i and κ_i $i = 1, \cdots, 102$, with respect to degree of each nodes in both layers. Each layer has 102 nodes. The IDN is a preferential attachment network with minimum node degree 20, and the contact network is a portion of a rural county social (face-to-face) network[20]. *Note:* Colors represent the optimum rates according to the colorbar. Nodes with identical node degree caused overlapping in the figure.

B.7 Conclusions

Based on the SAIS epidemic model, containment and suppression of an infectious disease spreading in a CN are possible with the help of disease-awareness diffusion among individuals. We proposed a method to optimally allocate available monetary resources for disease awareness. In particular, we determined optimal transition rates to a preventive-behavior state for each individual in the CN and IDN layers. We demonstrated that an epidemic can be contained in a multilayer network structure for a larger range of effective infection rates compared to a one-layer structure with the identical amount of resources. Furthermore, by allocating resources in both layers, epidemics can be contained that cannot be contained in a one-layer structure with more resources. Awareness rates are obtained by alternating the solution of a NPF problem and a convex optimization problem for an epidemic with a given effective infection rate until convergence is obtained. These optimum rates are positively correlated with node degrees in both layers. Therefore, any epidemic with identical or weaker effective infection rate is suppressed with a safety margin. This method selects the best individuals for adopting preventive behaviors with minimal costs and guaranteed epidemic dying-out.

Appendix C

Tutorial for GEMFPy: Generalized Epidemic Modeling Framework Software in Python

In this appendix, we are explaining the tutorial for the Python version of Generalized Epidemic Modeling Framework (GEMF) developed in[162]. For a more in depth report, please see[87;162]. We used NetworkX[163] in our examples due to its ubiquitous use for wrangling network data.

We are using compartmental epidemic models with networked contacts. For example, the SIS model has only two compartments: susceptible and infected. Transitions between compartments, are specified by the transition rates and their types. To determine nodal and edge-based transition rates, we use node transition graphs. However, for individual-based epidemic models, transition graphs represent only the transition mechanism for each node in the network and not for the entire population. The inducer compartment and layers that define neighbor nodes must also be specified. Other simulating paradigms, such as agent-based modeling[164–166] can follow the same methodology.

We present examples of epidemic models that can be simulated with GEMF. To implement the following code snippets, user should import GEMF with the following line[1].

```
from GEMFPy import *
```

C.1 SIS

Each node in an SIS model can be susceptible or infected; therefore, the number of compartments is $M = 2$. A susceptible node can become infected if it is surrounded by infected neighbors. Infection process of a node with one infected neighbor is a Poisson process with transition rate β. The infection processes are stochastically independent of each other; therefore, for a susceptible node with more than one infected node in its neighborhood, the transition rate is the infection rate β times the number of infected neighbor nodes. The neighborhood of each node is determined by a contact network. In addition to the infection process, a recovery process also exists. An infected node becomes susceptible again with a curing rate δ. The main characteristics and a node transition graph for the SIS model are shown in Table C.1 and Figure C.1.

Table C.1: Descriptions of the SIS model

SIS					
State	Transition	Type	Parameter	Inducer	Layer
S	$(S \rightarrow E)$	edge-based	β	Neighbors in I	1
I	$(I \rightarrow S)$	node-based	δ		

Parameters in Table C.1 can be entered by the following lines:

```
beta = 0.8; delta = 1;
Para = Para_SIS(delta,beta)
Net = NetCmbn([MyNet(G)])
```

where the function Para-SIS is defined as

```
def Para_SIS(delta,beta):
M = 2; q = np.array([1]); L = len(q);
```

[1] The latest version of GEMF can be found here.

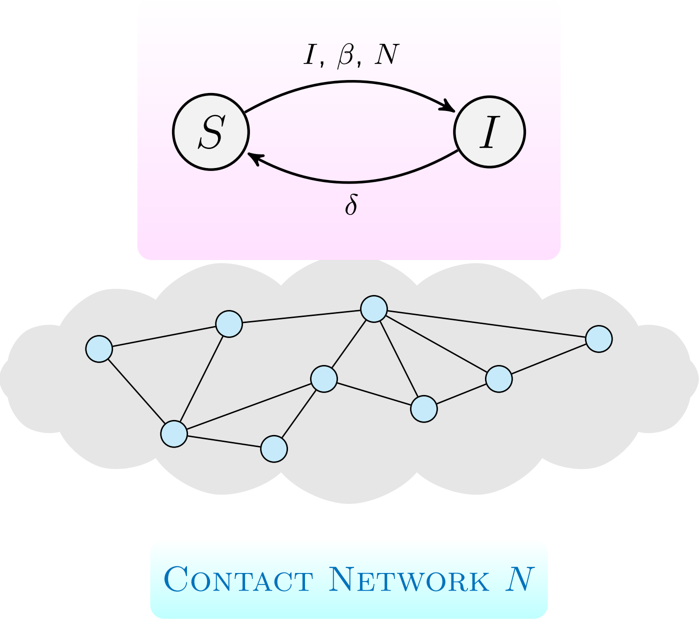


Figure C.1: Schematic of the network-based SIS model

```
A_d = np.zeros((M,M)); A_d[1][0] = delta
A_b = []
for l in range(L):
A_b.append(np.zeros((M,M)))
A_b[0][0][1] = beta #[l][M][M]
Para=[M,q,L,A_d,A_b]
return Para
```

we can choose initial condition such that two nodes are initially in the first inducer compartment[2] and others are in the first compartment:

```
x0 = np.zeros(N)
x0 = Initial_Cond_Gen(N, Para[1][0], 2, x0)
```

Simulation

After defining Para for SIS model, we simulated the SIS model with $\beta = 1.2$ and $\delta = 1$, as shown in Figure C.2. The simulation can be done by the following lines of codes: First

[2] Python is using 0-based indexing.

define the duration of simulation

```
StopCond = ['RunTime', 20]
```

and finding the occurred events:

```
ts, n_index, i_index, j_index = GEMF_SIM(Para, Net, x0, StopCond,n)
```

One output of the simulation can be the history for population of each compartment:

```
T, StateCount = Post_Population(x0, M, n, ts, i_index, j_index)
```

In Figure C.2, the fraction of nodes in each state is shown.

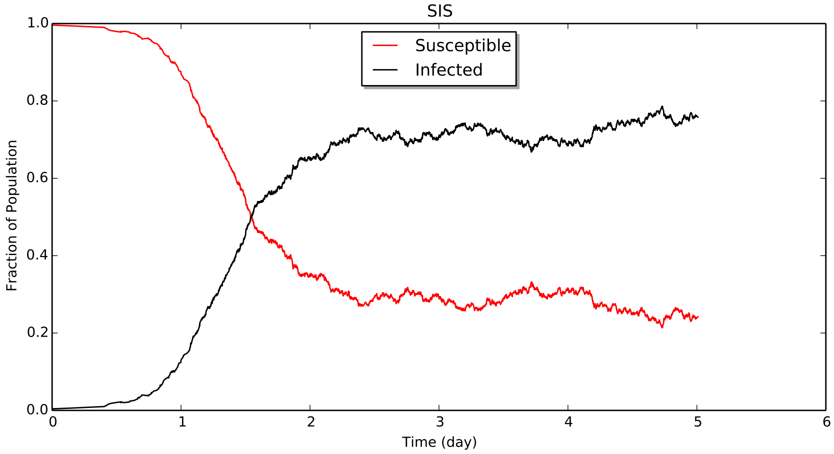


Figure C.2: Simulation of the SIS model

Due to stochastic nature of the simulation, we repeat it for multiple times (N) with the following snippet with a predefined step (see Section C.5.4):

```
N = 2
T_final = 3
step = .1
Init_inf = 2
t_interval, f = MonteCarlo(StopCond, Init_inf, M, T_final, step, N, n)
```

where n is the entire population size, step is the chosen time step and f is the averaged population of each compartment in the time step.

C.2 SIR

In the Susceptible-Infected-Recovered (SIR) model, each node can be either susceptible, infected, or recovered (immune). Therefore, the number of compartments, denoted by M,in the SIR model, was $M = 3$. A susceptible node can become infected if it is surrounded by infected nodes. The infection process of a node with one infected neighbor is a Poisson process with transition rate β. Similar to SIS, infection processes are stochastically independent of each other. In addition to the infection process, a recovery process also exists. An infected node recovers and becomes immune with a recovery rate δ. The main characteristics and a node transition graph for the SIR model are shown in Table C.2 and Figure C.3.

Table C.2: Descriptors of the SIR model

SIR multilayer					
State	Transition	Type	Parameter	Inducer	Layer
S	$(S \to E)$	edge-based	β	Neighbors in I	1
I	$(I \to R)$	node-based	δ		
R					

Parameters in Table C.2 can be entered by the following lines:

```
beta = 1.2; delta = 1;
Para = Para_SIR(delta, beta)
Net = NetCmbn([MyNet(G)])
```

where the function Para-SIR is defined as

```
def Para_SIR(delta, beta):
M = 3; q = np.array([1]); L = len(q);
A_d = np.zeros((M,M));   A_d[1][2] = delta
A_b = []
for l in range(L):
A_b.append(np.zeros((M,M)))
A_b[0][0][1] = beta #[l][M][M]
Para=[M,q,L,A_d,A_b]
return Para
```

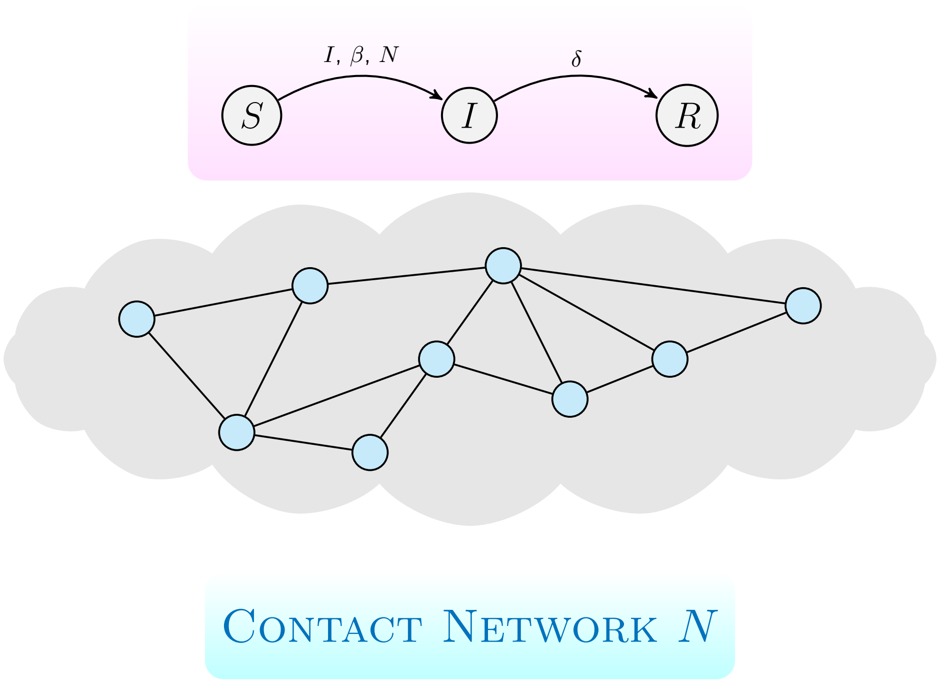


Figure C.3: Node transition graph for the SIR model for nodes in N

C.2.1 Simulation

After defining Para for SIR model, we simulated an SIR model with $\beta = 1.2$, $\delta = 1$, as shown in Figure C.3 for a Barabasi-Albert network with 500 nodes. Method is similar to SIS simulation in Section C.1.

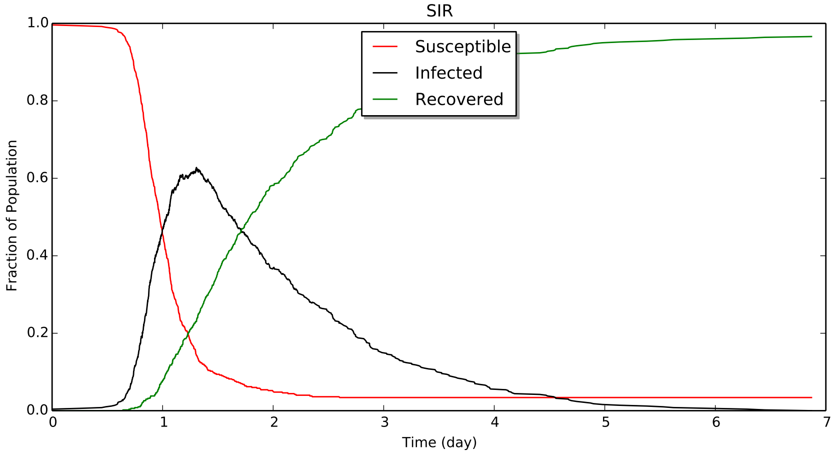


Figure C.4: Simulation of the SIR model

Table C.3: Descriptors of the SEIR

SEIR multilayer					
State	Transition	Type	Parameter	Inducer	Layer
S	$(S \rightarrow E)$	edge-based	β	Neighbors in I	1
E	$(E \rightarrow I)$	node-based	λ		
I	$(I \rightarrow R)$	node-based	δ		

C.2.2 SEIR

In the Susceptible-Exposed-Infected-Recovered (SEIR) model, each node can be susceptible, exposed, infected, or recovered (immune). Therefore, $M = 4$. A susceptible node can become exposed, if it is surrounded by infected nodes. The infection process of a node with one infected neighbor is a Poisson process with transition rate β. The neighborhood of each node is determined by a contact network N. An exposed node is not yet infectious, but it will transition to the infected state with rate λ. Finally, an infected node recovers with a recovery rate δ. The main characteristics and a node transition graph for the SEIR model are shown in Table C.3 and Figure C.5.

Parameters in Table C.3 can be entered by the following lines:

```
beta = 1.5; delta = 1; Lambda = .5
Para = Para_SEIR(delta, beta, Lambda)
Net = NetCmbn([MyNet(G)])
```

where the function Para-SEIR is defined as

```
def Para_SEIR(delta, beta, Lambda):
M = 4; q = np.array([2]); L = len(q);
A_d = np.zeros((M,M));   A_d[1][2] = Lambda; A_d[2][3] = Lambda
A_b = []
for l in range(L):
A_b.append(np.zeros((M,M)))
A_b[0][0][1] = beta #[l][M][M]
Para=[M,q,L,A_d,A_b]
return Para
```

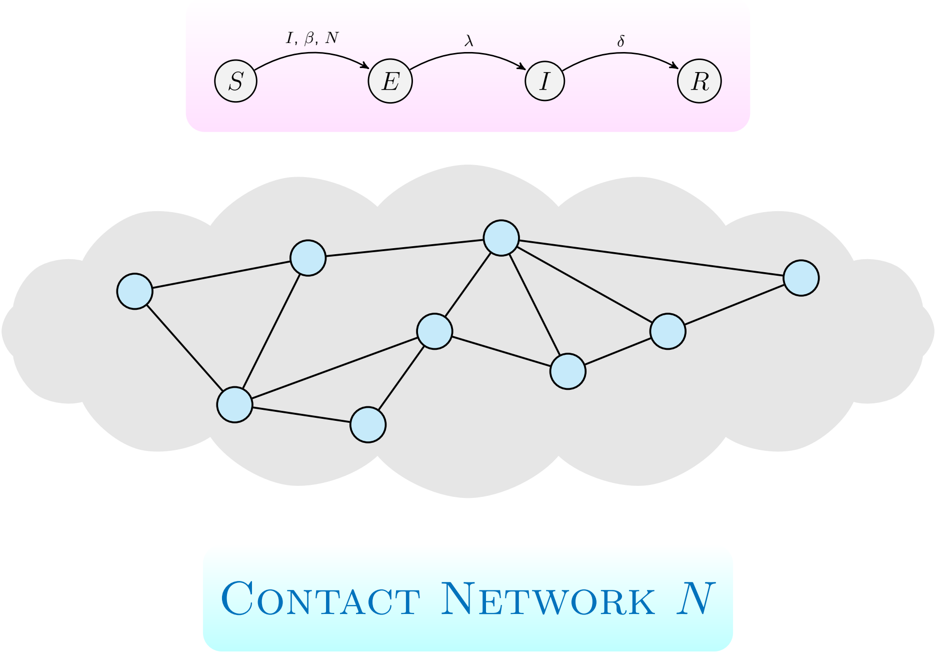


Figure C.5: Node transition graph for the SEIR model for nodes in N

C.2.3 Simulation

After defining Para for SEIR model, we simulated an SEIR model with $\beta = 1.2$, $\delta = 1$ and $\lambda = .4$, as shown in Figure C.6.

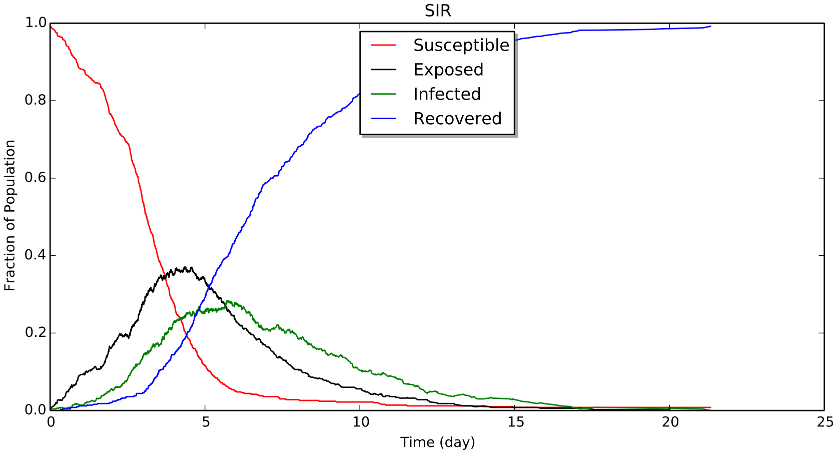


Figure C.6: Simulation of the SEIR model

C.3 SAIS

The Susceptible-Alert-Infected-Susceptible (SAIS) model was developed to incorporate individual reactions to the spread of a virus[148]. In the SAIS model, each node (individual) can be susceptible, infected, or susceptible-alert. Therefore, the number of compartments in the SAIS model was $M = 3$. The recovery process is similar to recovery process in the SIS model, characterized by the recovery rate δ. The infection process of a susceptible agent is also similar to the infection process of the SIS model, determined by infection rate β and contact network N. However, in the SAIS model, a susceptible node can become alert if it senses infected agents in its neighborhood. The alerting transition rate is κ times the number of infected agents. An alert node can also become infected by a process similar to the infection process of a susceptible node. However, the infection rate for alert nodes is lower than susceptible nodes due to the adoption of preventive behaviors. The alert infection rate is denoted by β_a with $0 < \beta_a < \beta$. The main characteristics and a schematic for the SAIS model are shown in the following Table C.4 and Figure C.7.

Table C.4: Descriptors of the SAIS single layer model.

SAIS single Layer					
State	Transition	Type	Parameter	Inducer	Layer
S	$(S \to I)$	edge-based	β	Neighbors in I	1
	$(S \to A)$	edge-based	κ	Neighbors in I	1
I	$(I \to S)$	node-based	δ		
A	$(A \to I)$	edge-based	β_a	Neighbors in I	1

Parameters in Table C.4 can be entered by the following lines:

```
Para = Para_SAIS_Single(delta, beta, beta_a, kappa)
```

where the function Para-SAIS for single layer is defined as

```
def Para_SAIS_Single(delta, beta, beta_a, kappa):
M = 3; q = np.array([1]); L = len(q);
A_d = np.zeros((M,M)); A_d[1][0] = delta
A_b = []
for l in range(L):
```

```
A_b.append(np.zeros((M,M)))
A_b[0][0][1] = beta #[l][M][M]
A_b[0][0][2] = kappa
A_b[0][2][1] = beta_a
Para = [M, q, L, A_d, A_b]
return Para
```

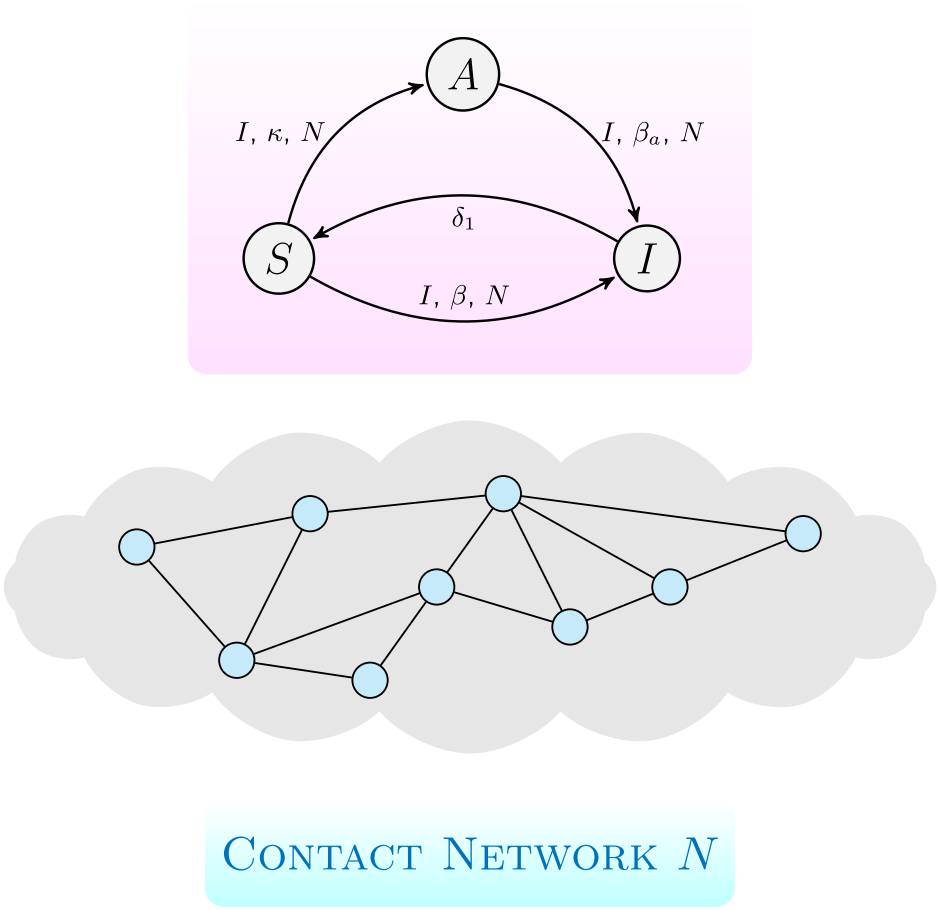


Figure C.7: Node transition graph for the SAIS one layer model for nodes in N

C.3.1 Simulation

After defining Para for SAIS model, we simulated an SAIS model in one-layer (N), with $\beta = \frac{5}{\lambda_1(\mathcal{G}_1)}$, $\delta = 1$ and $\beta_a = \frac{0.5}{\lambda_1(\mathcal{G}_1)}$, and $\kappa = 0.2\beta$, as shown in Figure C.8.

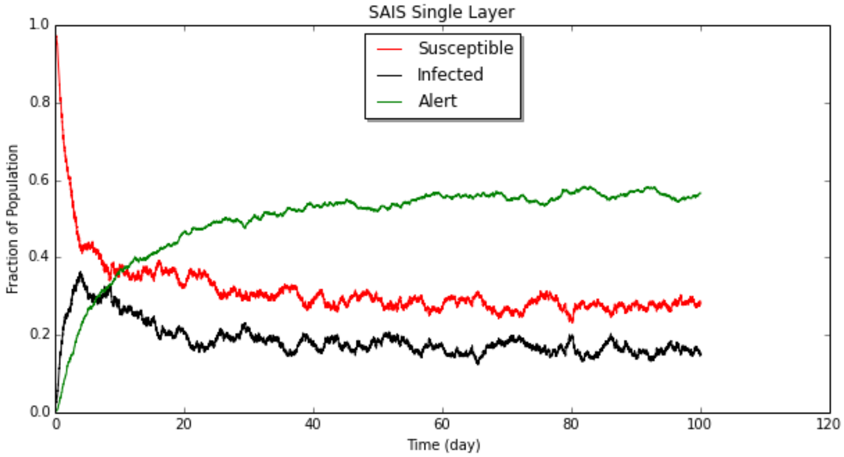


Figure C.8: Simulation of the SAIS single layer model.

C.3.2 SAIS Multilayer

The SAIS model on a two layer network was developed to incorporate multiple sources of information to react to the spread of the virus. In the SAIS spreading model, each node (individual) can be either susceptible, infected, or susceptible-alert. Again, the number of compartments in the SAIS model was $M = 3$. The infection process of a susceptible agent was also similar to the infection process of the SIS model, determined by infection rate β and contact network N_A. However, in this version of the SAIS model, a susceptible node can become alert if it senses infected agents in its contact neighborhood or if it is notified about infected neighbors in an information network N_B. The alerting transition rate is κ times the number of infected agents in the contact network and μ times the number of infected agents in the notification network. An alert node can also become infected by a process similar to the infection process of a susceptible node. However, the infection rate for alert nodes β_a is lower than β due to the adoption of preventive behaviors such as using masks. The main characteristics and a schematic for the SAIS-2 layer model are shown in Table C.5 and Figure C.10.

Parameters in Table C.5 can be entered by the following lines:

```
lambda1 = EIG1(G)[0]; delta = 1; beta = 5/lambda1; beta_a = .5/lambda1;
     kappa = .2*beta; mu = .5*beta
```

Table C.5: Descriptors of the SAIS two-layer model

SAIS multilayer					
State	Transition	Type	Parameter	Inducer	Layer
S	$(S \to I)$	edge-based	β	Neighbors in I	1
	$(S \to A)$	edge-based	κ	Neighbors in I	1
	$(S \to A)$	edge-based	μ	Neighbors in I	2
I	$(I \to S)$	node-based	δ		
A	$(A \to I)$	edge-based	β_a	Neighbors in I	1

```
Para = Para_SAIS(delta, beta, beta_a, kappa, mu)
```

where the function Para-SAIS for two layer is defined as

```
def Para_SAIS(delta, beta, beta_a, kappa, mu):
M = 3; q = np.array([1,1]); L = len(q);
A_d = np.zeros((M,M)); A_d[1][0] = delta
A_b = []
for l in range(L):
A_b.append(np.zeros((M,M)))
A_b[0][0][1] = beta #[l][M][M]
A_b[0][0][2] = kappa
A_b[1][2][1] = beta_a
A_b[1][0][2] = mu
Para = [M, q, L, A_d, A_b]
return Para
```

C.3.3 Simulation

After defining Para for SAIS model, we simulated the process with $\beta = \frac{5}{\lambda_1(\mathcal{G}_1)}$, $\delta = 1$ and $\beta_a = \frac{0.5}{\lambda_1(\mathcal{G}_1)}$, $\kappa = 0.2\beta$, and $\mu = 0.5\beta$, as shown in Figure C.11.

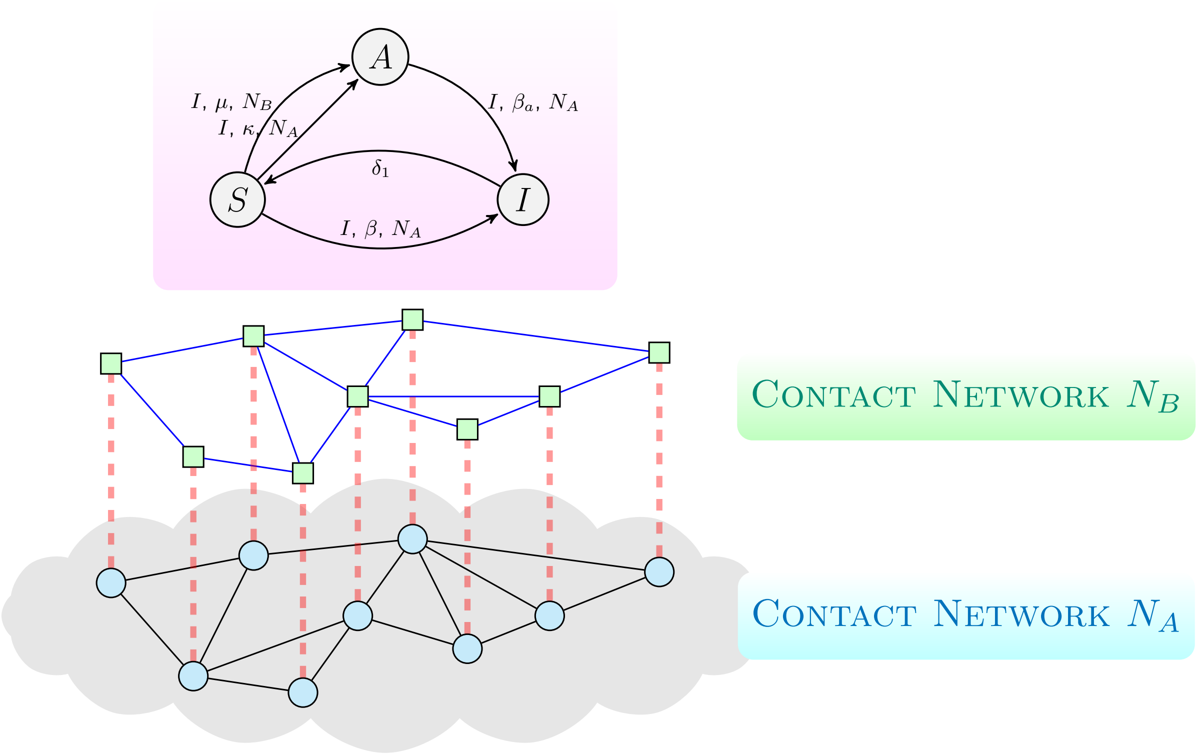


Figure C.9

Figure C.10: Node transition graph for the SAIS two-layer model on network with layers N_A and N_B.

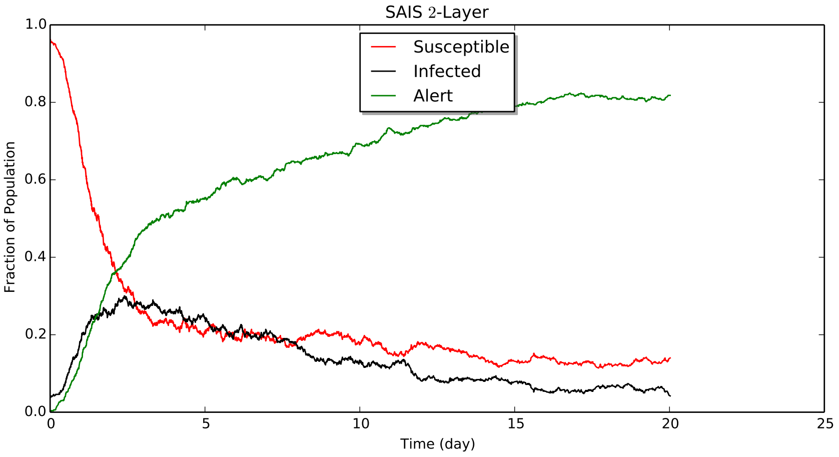


Figure C.11: Simulation of the SAIS in 2 layer

C.4 Multiple interacting pathogen spreading $\text{SI}_1\text{SI}_2\text{S}$

Assigning only one influencer compartment to one network layer allows different elegant analysis. However, a more general possibility is that an edge-based transition $m \to n$ occurs if a neighbor j, is in a subset of the compartments, such as $q_{l,1}$ or $q_{l,2}$. This case can be treated within the same structure, allowing the network layer to be counted twice. For example, we assumed that in the first layer the model had the influencer compartment $q_{l,1}$, and in the second layer, the graph has the influencer compartment $q_{l,2}$.

The $\text{SI}_1\text{SI}_2\text{S}$ model is an extension of continuous-time SIS spreading of a single virus on a simple graph, to the modeling of competitive viruses on a two-layer network[167]. In this model, each node is either susceptible, 1-infected, or 2-infected (i.e., infected by Virus 1 or 2, respectively). Virus 1 spreads through network N_1, virus 2 spreads through network N_2. In this competitive scenario, the two viruses are exclusive: a node cannot be infected by Virus 1 and Virus 2 simultaneously. Consistent with SIS propagation on a single layer, the infection and recovery processes for Virus 1 and 2 have similar characteristics. The curing process for 1-infected Node i is a Poisson process with recovery rate $\delta_1 > 0$. The infection process for susceptible Node i effectively occurs at rate $\beta_i Y_i(t)$, where $Y_i(t)$ is the number of 1-infected neighbors of node i at time t in layer N_1. Recovery and infection processes for Vvirus 2 are similarly described. The main characteristics and a node transition graph for the $\text{SI}_1\text{SI}_2\text{S}$ model are shown in Table C.6 and Figure C.12.

Table C.6: Descriptions of the $\text{SI}_1\text{SI}_2\text{S}$ model. S: suscpetible, I_1: infected by virus 1, I_2: infected by virus 2,

$\text{SI}_1\text{SI}_2\text{S}$					
State	Transition	Type	Parameter	Inducer	Layer
S	$(S \to I_1)$ $(S \to I_2)$	edge-based edge-based	β_1 β_2	Neighbors in I_2 Neighbors in I_2	1 2
I_1	$(I_1 \to S)$	node-based	δ_1		
I_1	$(I_2 \to S)$	node-based	δ_2		

Parameters in Table C.6 can be entered by the following lines in two different networks N1 (G) and N2 (H):

```
N = G.number_of_nodes()
lambda1_1 = EIG1(G)[0]; lambda1_2 = EIG1(H)[0]; delta1 = 1; beta1 = 5/lambda1_1;  delta2 = 1; beta2 = 5/lambda1_2;
Para = Para_SI1I2S(delta1, delta2, beta1, beta2)
Net = NetCmbn([MyNet(G), MyNet(H)])
x0 = np.zeros(N)
x0 = Initial_Cond_Gen(N, Para[1][0], 20, x0)
x0 = Initial_Cond_Gen(N, Para[1][1], 20, x0)
```

where the function Para-SI$_1$SI$_2$S is defined as

```
def Para_SI1I2S(delta1, delta2, beta1, beta2):
M = 3; q = np.array([1,2]); L = len(q);
A_d = np.zeros((M,M)); A_d[1][0] = delta1; A_d[2][0] = delta2
A_b = []
for l in range(L):
A_b.append(np.zeros((M,M)))
A_b[0][0][1] = beta1 #[l][M][M]
A_b[1][0][2] = beta2 #[l][M][M]

Para = [M, q, L, A_d, A_b]
return Para
```

C.4.1 Simulation

After defining Para for this model, we simulate an S$_1$SI$_2$S model in one layer with $\beta_1 = \frac{5}{\lambda_1(\mathcal{G}_1)}$, $\beta_2 = \frac{5}{\lambda_1(\mathcal{H}_1)}$, $\delta_1 = 1$ and $\delta_2 = 1$ in Figure C.13.

C.5 An overview of the functions

In the subsequent sections, we describe the following functions of Gemf:

1. Initialization functions

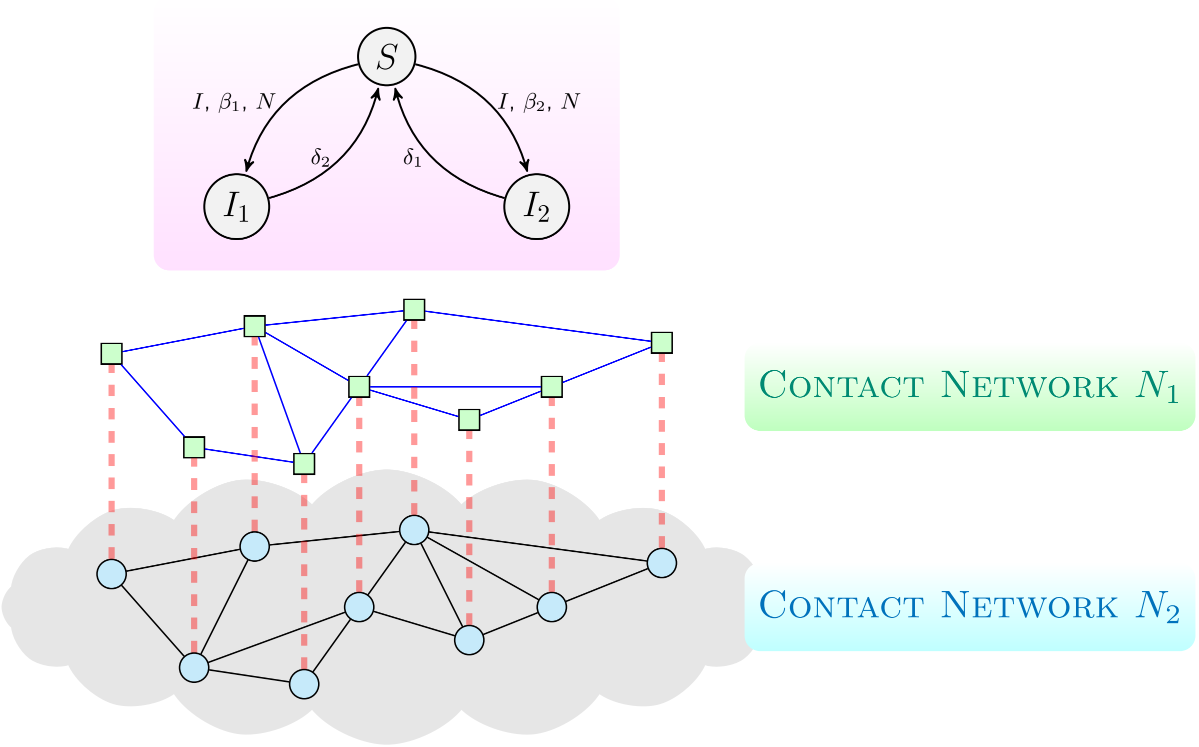


Figure C.12: Node transition graph for the SI_1SI_2S in two layer model

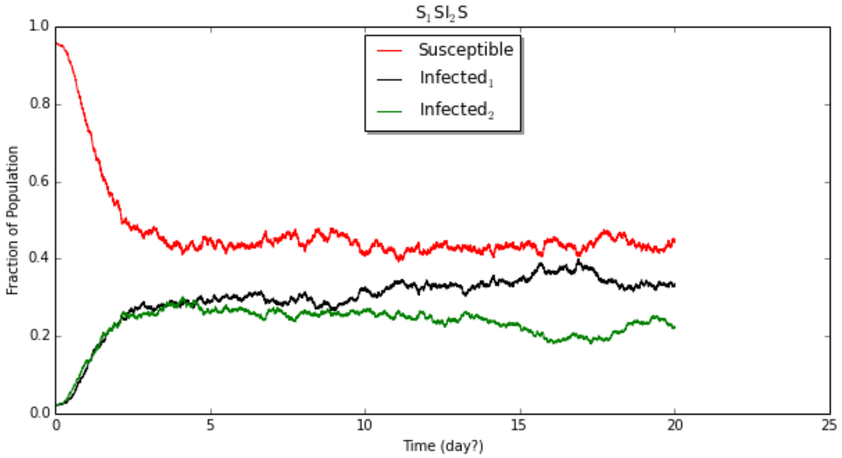


Figure C.13: Simulation of the S_1SI_2S model

(a) Net

(b) NetComb

(c) Para

   (d) InitialCond

2. Simulations

   (a) Sim

   (b) PostProcess

   (c) MonteCarlo

3. Output

   (a) Visualization

C.5.1 Initialization

“Net”: Converting network data

Gemf converts graph information into graph adjacency list format with function Net; therefore, we recorded i, j, and w in vectors L_2, L_1 and W for each edge (i, j, w) . L_2 are neighbors of L_1 with weight W, and we sorted L_2 and re-arranged L_1 and W with resulting sorted arguments in order to organize these data. All network data was acquired with L_1, L_2 and W.

The NeighborhoodData function was used, which takes L_1 and L_2 as its inputs and returns 5 vectors outputs. *Neighvec* and *NeighWeight* are vectors of neighbors, and their weights I_1 and I_2, respectively, are node indices and d is their edge multiplicity. Benefits of this representation are described in Section C.5.2.

We defined NNeighborhoodData in order to distinguish between neighbors of node i with nodes that have i as neighbors. This function, is useful when we are dealing with a directed network and has the same structure as the previous function. Outputs of this function, *NNeighvec* and *NNeighWeight*, are vectors of adjacent nodes (not neighbors) and their edge weights respectively and NI_1 and NI_2 are indices and Nd is edge multiplicity. Function Net returns all above information for a single layer.

Combining network layers data

For each layer, we obtained the required information from Net, and we combined them with the NetComb function.

$$\begin{aligned}\textsc{NetComb}\left(\{Net_1, \cdots, Net_L\}\right) = \\ \left[\left[Net_1\left(1\right), \cdots, Net_L\left(1\right)\right], \cdots, \left[Net_1\left(H\right), \cdots, Net_L\left(H\right)\right]\right]_{1\times L}\end{aligned} \tag{C.1}$$

where $Net_l = \textsc{Net}\left(\mathcal{G}_l\right)$.

Transition rates

We used Para function to enter the required data for transition rates. A nodal transition rate matrix is an $M \times M$ matrix in which entry mn represents the rate of nodal transition $m \to n$:

$$A_\delta \triangleq \left[\delta_{mn}\right]_{M\times M}. \tag{C.2}$$

An edge-based transition rate matrix, corresponding to the network layer l, is an $M \times M$ matrix in which entry mn represents the rate $\beta_{l,mn} > 0$ of edge-based transition $m \to n$:

$$A_\beta \triangleq \left[\left[\beta_{1,mn}\right]_{M\times M}, \cdots, \left[\beta_{L,mn}\right]_{M\times M}\right]_{1\times L}. \tag{C.3}$$

Initial condition

With "InitialCondGen" function the initial status of each individual in the population can be determined and various approaches can be used to do this.

- User input: Initial condition is directly chosen by the user.
- Fixed initial infected population: N_J individuals randomly chosen to be in compartment J.

C.5.2 Simulations

GEMF uses an event-driven approach to simulate the stochastic process. This method is advantageous compared to the discretized method. For example, in discretization approach, no transition may occur in several time increments dt or several transition may occur in one time increment; therefore, computation time for the event-based method is not unnecessarily longer and on the other side the solution is more accurate and captures more events compared to the discretized method (See[168;169]).

Number of neighbors in influencer compartment N_q

One of the key factors in edge-based transitions is the number of neighbors in influencer compartment, N_q. N_q is an $L \times N$ array, representing the number of influencer compartment for each node in each layer, weighted by edge weights. Because node status changes in each event, N_q is updated after each event. From Section C.5.1, initial status of all nodes $X^0_{M\times N}$ is obtained. For example, if $X\left[:,4\right]^T = \begin{bmatrix} 0 & 1 & \cdots & 0 \end{bmatrix}_{1\times M}$, then node 4 is in compartment 2.

To compute N_q, GEMF goes over all nodes in each layer. Using network data from NETCOMB, all neighbors of node n in layer l can be derived via

$$N_{ln} = Neigh\left[l\right]\left[I_1\left[l,n\right] : I_2\left[l,n\right]\right] \tag{C.4}$$

with weights:

$$W_{ln} = NeighWeight\left[l\right]\left[I_1\left[l,n\right] : I_2\left[l,n\right]\right]. \tag{C.5}$$

Using (C.5), entries of N_q (influencer neighbors) can be determined by

$$N_q\left[l,n\right] = \sum_{i=1}^{|N_{ln}|} X\left[q\left[l\right], N_{ln}\left[i\right]\right] \cdot W_{ln}\left[i\right] \tag{C.6}$$

where $|N_{ln}|$ is the cardinality of set N_{ln}.

Rate of changes

From Section C.5.1, we entered A_β and A_δ through Para. The simulation code initially generated b_{il}, which is an arrays:

$$b_{il} \triangleq \begin{bmatrix} \begin{bmatrix} \sum_{i=1}^{M} \beta_{1,1i} \\ \vdots \\ \sum_{i=1}^{M} \beta_{1,Mi} \end{bmatrix}_{M\times 1} & \cdots & \begin{bmatrix} \sum_{i=1}^{M} \beta_{L,1i} \\ \vdots \\ \sum_{i=1}^{M} \beta_{L,Mi} \end{bmatrix}_{M\times 1} \end{bmatrix}_{1\times L} \tag{C.7}$$

where b_{il} represents the sum of edge-based transition rates of each compartment in each layer.

The array of edge-based transition rates matrix for each compartment in all layers b_i was

$$b_i \triangleq \begin{bmatrix} \begin{bmatrix} \beta_{1,11} & \cdots & \beta_{L,11} \\ \beta_{1,12} & \cdots & \beta_{L,12} \\ \vdots & \ddots & \vdots \\ \beta_{1,1M} & \cdots & \beta_{L,1M} \end{bmatrix}_{M\times L} & \cdots & \begin{bmatrix} \beta_{1,21} & \cdots & \beta_{L,21} \\ \beta_{1,22} & \cdots & \beta_{L,22} \\ \vdots & \ddots & \vdots \\ \beta_{1,2M} & \cdots & \beta_{L,2M} \end{bmatrix}_{M\times L} \end{bmatrix}_{1\times M} \tag{C.8}$$

For each compartment, the total leaving rate due to nodal transition was derived from C.2 (by summing up each row of matrix A_δ):

$$d_i = \begin{bmatrix} \sum_{i=1}^{M} \delta_{1i} \\ \vdots \\ \sum_{i=1}^{M} \delta_{Mi} \end{bmatrix}_{M\times 1} . \tag{C.9}$$

Total Rates

Using d_i and b_{il}, total transition rates for each node were generated as

$$R_{in} = (d_i \mathbf{1}_{1\times N})_{M\times N} \circ X + (b_{il} N_q)_{M\times N} \circ X \tag{C.10}$$

where $\circ$ represents element-wise multiplication.

In order to find the total rate of change for the entire system, we re-added the rates. For example, for the total rate of change for each compartment in the entire network, we introduce R_i:

$$R_i = \begin{bmatrix} \sum_{i=1}^{N} R_{in}\left[1, i\right] \\ \vdots \\ \sum_{i=1}^{N} R_{in}\left[M, i\right] \end{bmatrix}_{M \times 1} \tag{C.11}$$

and for the total rate of change for the entire system, we introduced R:

$$R = \sum_{i=1}^{M} R_i\left[i\right]. \tag{C.12}$$

Updating system status after an event

The initial state for all nodes was generated according to Section C.5.1. Because all random processes are Poisson processes, the assumption was made that the next event would occur in time δt:

$$\delta t = \frac{-\ln(\mathsf{rand})}{R} \tag{C.13}$$

where $0 \leq \mathsf{rand} \leq 1$ is a generated random number. During this event one of the nodes changes its status. We determined which compartment changed by drawing a sample among M compartments with probability distribution R_i; this compartment was called i_s.

Once the leaving compartment was identified, we wanted to know which node experienced the transition. Therefore, we drew a sample from N nodes with probability distribution $R_in\left[i_s, :\right]$ (i.e., i_s row of matrix R_in) and called this Node n_s.

To find the new status (compartment) of Node n_s, again GEMF randomly draws the new compartment j_s among M compartments with the following probability distribution:

$$p_{j_s}^T = A_\delta\left[i_s, :\right]^T + b_i\left[i_s\right] N_q\left[:, n_s\right]. \tag{C.14}$$

Drawing samples with given probability distribution is done with RndDraw function.

With δt, i_s, j_s, and n_s, Gemf had all necessary information to update the network status and apply required changes with the occurred event. However, Gemf had to update X matrix and the future rate of transitions.

Because Node n_s changed its status from i_s to j_s, we have:

$$X\left[i_s, n_s\right] = 0,\ X\left[j_s, n_s\right] = 1. \tag{C.15}$$

To update R_i, we subtracted the column in R_{in} that corresponded to Node n_s (i.e., $R_{in}\left[:, ns\right]$) and then we updated

$$R_{in}\left[:, n_s\right] = d_i \circ X\left[:, n_s\right] + \left(b_i N_q\left[:, n_s\right]\right) \circ X\left[:, n_s\right]. \tag{C.16}$$

Now we add $R_{in}\left[:, n_s\right]$ to R_i. Next if any of the old or new compartment are in influencer category in any layer, code should update N_q matrix. First, we find neighbors of node n_s:

$$N_{ln} = Neigh\left[l\right]\left[I_1\left[l\right]\left[n_s\right] : I_2\left[l\right]\left[n_s\right]\right] \tag{C.17}$$

$$WeightedNeigh = NeighW\left[l\right]\left[I_1\left[l\right]\left[n_s\right] : I_2\left[l\right]\left[n_s\right]\right] \tag{C.18}$$

$$\tag{C.19}$$

Then we conducted the following steps for all these neighbors:

- If the old compartment i_s was an influencer compartment in layer l, we do the following removed n_s as their infected neighbors and recorded the weight of the edge. We also updated R_{in}. For n, the k'th neighbor of n_s was

$$N_q\left[l\right]\left[n\right] -= NNeighW\left[l\right]\left[NI_1\left[l\right]\left[n_s\right] + k\right] \tag{C.20}$$

$$R_{in}\left[:, n\right] -= NNeighW\left[l\right]\left[NI_1\left[l\right]\left[n_s\right] + k\right]\left(b_{il}\left[:, n\right] \circ X\left[:, n\right]\right) \tag{C.21}$$

where $-=$ indicates subtracting to current value.

- If the new compartment j_s was an influencer compartment in layer l, we added n_s as their infected neighbors and recorded the weight of the edge. We also updated R_{in}. For n, the kth neighbor of n_s was

$$N_q\left[l\right]\left[n\right]+=NNeighW\left[l\right]\left[NI_1\left[l\right]\left[n_s\right]+k\right] \tag{C.22}$$

$$R_{in}\left[:,n\right]+=NNeighW\left[l\right]\left[NI_1\left[l\right]\left[n_s\right]+k\right]\left(b_{il}\left[:,n\right]\circ X\left[:,n\right]\right) \tag{C.23}$$

where $+=$ indicates adding to current value.

We stacked n_s, j_s, and i_s into n_{index}, j_{index}, and i_{index}, respectively, and then we recalculated R_i and R and prepared for the next event.

C.5.3 Post processing

From Sim, we obtained the set of time increments of occurring events, t_s. The cumulative sum of t_s, T, was the time history of events. $StateCount$, an $M \times (|t_s|+1)$ array conveying the total number of nodes in each compartment in each time step, , was also generated. The First column of $StateCount$ is initial condition:

$$StateCount\left[:,1\right]=\sum_{i=1}^{N} X_0\left[:,i\right]. \tag{C.24}$$

For the remainder of $StateCount$, PostPopulation generated a temporary array $DX_{M\times 1}$ in each event, with the following non-zero elements:

$$DX\left[i_{index}\left[k\right]\right]=-1 \tag{C.25}$$

$$DX\left[j_{index}\left[k\right]\right]=1, \tag{C.26}$$

and zero on the other elements. In each event, we obtained the following recursion:

$$StateCount\left[:,k+1\right] = StateCount\left[:,k\right] + DX \tag{C.27}$$

PostPopulation returns T and $StateCount$.

C.5.4 Monte carlo simulation

In order to obtain a reliable result for stochastic simulation, it is necessary to repeat random processes had to be repeated for many times and the results need to be averaged.

In an event-based analysis, the number of events are not identicale for different simulations; therefore, arrays that show the state of the group in each simulation will not be of the same size and they cannot be added and averaged.

In order to average several random processes, a ubiquitous time interval with a desired time increment must be defined.

Therefore, the function MonteCarlo, uses histogram counting. For all simulations, it finds the closest previous event for the time increments and then maps these events on the new time interval. With the new unified time scale, the average of all processes was derived.