Markov Tensor Theory and Cascade, Reachability, and Routing in Complex Networks

A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Golshan Golnari

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Zhi-Li Zhang

December, 2017

Acknowledgements

There are many people that have earned my gratitude for their contribution to my time in graduate school. First and foremost, I would like to thank my adviser, Professor Zhi-Li Zhang, for his continuous support and guidance on conducting research and building a strong and reliable skill set throughout my PhD career. The work presented in this dissertation would not have happened without his patience and motivation.

I would like to thank professors Daniel Boley, Soheil Mohajer, Nikolaos Sidiropoulos, Dmytro Bilyk, and Andrew Lamperski for serving on my proposal and final thesis defense. Their feedback and guidance have been absolutely invaluable. My sincere thank goes to Professor Banerjee who was the first to motivate me to enter the field of machine learning and data science and I owe a major part of my career success to his classes and the discussions I had with him and his student, Amir Asiaei, over a paper together. Moreover, I want to thank all the professors who gave me the opportunity to learn and flourish; Daniel Boley, Tom Luo, Pavlo Pylyavskyy, Shashi Shekhar, Ravi janardan, Anand Gopinath, Paul Schrater, David Du, Stergios Roumeliotis, Rui Kuang, and Tian He.

I am grateful to my collaborators and my labmates whom I learned a lot from and enjoyed prolonged technical discussions with; Amir Asiaei, Yanhua Li, Braulio Dumba, Gyan Ranjan, Eman Ramadan, and Cheng Jin. I also enjoyed the companionship of many friends during my years in grad school, Fatemeh (kh), Fatemeh (F), Fatemeh (A), Fatemeh (S), Pantea, Leila, Roushanak, Zahra, Yasaman, Ameneh, Sareh, Narges, Atefeh, and Zeinab.

Most importantly, I would like to thank my family for their unconditional support and love. My parents, Rahim and Zahra, always had the highest expectations for our education and they devoted their life for us to peruse our enthusiasm for learning and

to achieve the best. I would like to express my greatest gratitude to my parents and my mother-in-law, Mahboubeh, without whose visit and help during the last year of my PhD, I was not able to wrap up my research after having a new born and working as a full-time scientist at 3M research lab. Special thanks to my siblings, Pareesa, Pedram, and Golnaz whose presence and companion in the important events of life have been always heart warming. Undoubtedly, their presence in my graduation ceremony alongside my parents has been one of the most memorable moments in my lifetime. I would also like to thank my siblings-in-law, Khatoon, Saleh, and Sadegh for their kindness and support. I wish I could share my happiness of graduation with my father-in-law, Ali, whose memory is always with us.

The words are impotent to thank my love, my husband, my friend, my classmate, my colleague, my advisor, my supporter, and my pillar of strengths who all appeared in one person, Saber, without whose encouragement, patience, and advice I would never have started and completed this journey.

At the end I should say, mamnoonam ey khoda baraye hame lotf va na'amatet.

Dedication

To the love of my life, Saber,
and to the blossom of our love, Ala.

Abstract

In this dissertation, we study and characterize the networks as the medium and substrate for communications, interactions, and flows by addressing various crucial problems under the general topics of *cascade*, *reachability*, and *routing*. These are general problem domains common in several applications and from a variety of networks. We address these problems in a unified way by a theoretical platform that we have developed in this research, which we call *Markov Tensor Theory*.

How does a phenomena, influence, or a failure cascade in a network and what are the key factors in this cascade? We study the influence cascade in social networks and introduce the Heat Conduction (HC) Model which captures both social influence and non-social influence, and extends many of the existing non-progressive models. We then prove that selecting the optimal seed set of influential nodes for maximizing the influence spread is NP-hard for HC, however, by establishing the submodularity of influence spread, we tackle the influence maximization problem with a scalable and provably near-optimal greedy algorithm. We also study failure cascade in inter-dependent networks where we considered the effects of cascading failures both within and across different layers. In this study, we investigate how different couplings (i.e., inter-dependencies) between network elements across layers affect the cascading failure dynamics.

How failures or disruptions affect the network in terms of reachability of entities from each other, how to identify the reachabilities efficiently after failures, and who are the pivotal players in the reachabilities? We develop an oracle to answer dynamic reachabilities efficiently for failure-prone networks with frequent reachability query requirement. Founded on the concept of reachability, we also introduce and provide a formulation for finding articulation points, measuring network load balancing, and computing pivotality ranking of nodes.

Once the reachabilities are determined, how to quickly and robustly route a flow from a part of network to the other part of network under the failures? To avoid solely relying on the shortest path and generate alternative paths on one hand, and to correct the degeneracy of hitting time distance on the other hand, we develop a novel routing continuum method from shortest-path routing to all-path routing which provides both a

closed form formulation for computing the continuum distances and an efficient routing strategy. We also devise an oracle for efficiently answering to single-source shortest path queries as well as finding the replacement paths in the case of multiple failures.

For these studies, we develop *Markov Tensor Theory* as a platform of powerful theories and tools founded on Markov chain theory and random walk methods which supports the general *weighted* and *directed* networks.

Contents

List of Tables

List of Figures

Part I

Introduction

The world is surrounded by an immense variety of complex networks; from infrastructure networks which are more recognizable as *network*, such as power grid, water distribution network, computer networks, and Internet, to interpretative networks whose links represent some form of relationships and interactions rather than real physical means, such as protein-protein interaction networks in biochemistry, networks of flights and airports, social networks in online social media like Facebook, and cause-and-effect networks in our daily life where the nodes model the events and links are the causality relations between the events. In all these sorts of networks, the essence of network is to play as a *medium* for connectivity of entities such that they can communicate and interact with each other and to have units of influence *flow* from some parts of the network to other parts in the network.

In this dissertation, we study and characterize the networks as the medium and substrate for communications, interactions, and flows by addressing various crucial questions and problems under the general topics of *cascade*, *reachability*, and *routing*: How does a phenomena, influence, or a failure *cascade* in a network and what are the key factors in this cascade? How failures or disruptions affect the network in terms of *reachability* of entities from each other, how to identify the reachabilities efficiently after failures, and who are the pivotal players in the reachabilities? Once the reachabilties are determined, how to quickly and robustly *route* a flow from a part of network to the other part of network under the failures?

To address such questions, we have developed *Markov Tensor Theory* which is a platform of powerful theories and tools founded on Markov chain theory and random walk methods, and supports the general *weighted* and *directed* networks.

In the following, we provide the outline of thesis: Markov Tensor Theory is briefly discussed in 0.1 and the applications of these theories are overviewed in Sections 0.2,0.3,0.4. The related work is presented later in each related chapter.

0.1 Markov Tensor Theory

Markov Tensor Theory is a layered-structure initiated from *fundamental tensor*, advanced to the next layer of *avoidance fundamental tensor*, and completed by the last layer of *avoidance fundamental tensor in evaporation paradigm*. By adding each layer,

we provide more flexibility and creativity in designing the Markov chain metrics to capture and model more advanced cases in network problems. Markov Tensor Theory is also a unified basis for generating the other Markov chain metrics, including hitting time and absorption probability, which have been also shown to be very effective in network analysis. We establish the effectiveness of Markov Tensor Theory in finding the most influential people in a social network for influence maximization 0.2.1, devising an oracle to efficiently answer dynamic reachability queries 0.3, computing the articulation points of directed networks 0.3, proposing a pivotality metric to rank the importance of nodes in reachabilities 0.3, developing a generative model for a routing continuum from shortest path to (random walk) all path 0.4.1, and devising a distance oracle which answers to single-source shortest path (SSSP) queries 0.4.2 and finds replacement paths in multiple failures efficiently0.4.3.

Random walk methods and Markov chain, which are in close relationship, shown to be powerful tools in many fields from physics and chemistry to social sciences, economics, and computer science [80, 81, 136, 1, 23]. For network analysis, too, they have shown promises as effective tools [99, 29, 121, 50, 24, 79]. Chapter 1 provides the preliminary background on Markov chain, harmonic functions, and networks.

In Chapter 2, we review certain Markov chain classical metrics, including fundamental matrix, hitting time, hitting cost, and hitting (absorption) probability, and their diverse forms of definitions under a respective and consonant framework. We reveal and prove theoretically the connections between these metrics as well as the connections between their different definitions. We propose the fundamental tensor as a generalization of fundamental matrix and show how all the mentioned metrics can be computed from the fundamental tensor in a unified way. We also review and prove a library of useful relations, lemmas, and theorems on the Markov chain classical metrics.

In Chapter 3, we develop Markov chain *avoidance metrics* as an extension of Markov chain classical metrics. While the Markov chain classical metrics are the results of imposing only the stopping criteria, i.e. hitting the target state for the first time, and has no control or conditions on the visiting states in the middle of the transition, the Markov chain avoidance metrics provide more flexibility in the design of Markov chain and impose new conditions on the transition to avoid (or transit) a specific state (or a set of states) before the stopping criteria. In particular, we introduce avoidance fundamental tensor,

avoidance hitting time, transit hitting time, and avoidance hitting cost and establish that all of them can be computed from the fundamental matrices associated with the appropriately defined transition probability matrices. We also introduce the evaporation paradigm in this chapter, but defer the complete definition of avoidance fundamental tensor in evaporation paradigm to Chapter 7 where the related application makes it easier to understand the metric.

0.2 Cascade

We are witnessing cascades happening all over real networks when a phenomena is initiated from a part of network and propagates to the rest of network. This phenomena could be a news or publicity of a product where people are influencing each other toward the promotion of this cascade, or could be a failure of entities where failing in functionality of some entities can cause further failures in the network. In this dissertation, we study the cascade of influence and the cascade of failure in networks, in the context of social networks and cyber-physical interdependent networks respectively 0.2.10.2.2.

0.2.1 Influence Cascade

A social network plays a key role as a medium for the spread of information, ideas, and influence among its members. The Influence maximization problem is about finding the most influential persons who can maximize the spread of influence in the network. This problem has applications in viral marketing where a company may wish to spread the publicity and eventually the adoption of a new product via the most influential persons. A social network is modeled by a graph where nodes represent the users, and edges represent relationships and interactions between the users. An influence cascade over a network can be modeled by a diffusion process, and the objective of influence maximization problem is to find k most influential persons as the initial adopters who will lead to largest number of adoptions.

In Chapter 4, we first propose the Heat Conduction (HC) model as a diffusion process which has favorable real-world interpretations and can capture the *reversibility of choices*. We then prove that selecting the optimal seed set of influential nodes is NP-hard for HC, however, by establishing the submodularity of influence spread, we can tackle

the influence maximization problem with a scalable and provably near-optimal greedy algorithm. In sharp contrast to the other greedy influence maximization methods, our fast and efficient C2Greedy algorithm benefits from two analytically computable steps: *closed-form* computation for finding the influence spread and the greedy seed selection. Through extensive experiments on several and large real and synthetic networks, we show that C2Greedy outperforms the state-of-the-art methods, under HC model, in terms of both influence spread and scalability.

0.2.2 Failure Cascade

Many real-world (cyber-)physical infrastructure systems are multi-layered, consisting of multiple *inter-dependent* networks/layers. Due to this interdependency, the failure cascade can be catastrophic in an inter-dependent multi-layered system and even leads to the break-down of the entire system. The 2003 blackout of the Italian power grid was reportedly the result of a cascading failure due to the inter-dependency of the power grid and the communication network that it relied on [114].

In Chapter 5, we propose a theoretical framework for studying cascading failures in an inter-dependent multi-layered system where we consider the effects of cascading failures both *within* and *across* different layers. The goal of the study is to investigate how different *couplings* (i.e., inter-dependencies) between network elements across layers affect the cascading failure dynamics. Through experiments using the proposed framework, we show that under the one-to-one coupling, how nodes from two inter-dependent networks that are coupled together play a crucial role in the final size of the resulting failure cascades: coupling corresponding nodes from two networks with equal importance (i.e., "high-to-high" coupling) result in smaller failure cascades than other forms of inter-dependence coupling such as "random" or "low-to-high" coupling. Our results shed lights on potential strategies for mitigating cascading failures in inter-dependent networks.

0.3 Reachability

Network reachability analysis is of great importance for several applications [129, 26, 137, 75, 104]. When it is said that node v is reachable from node u, it means that there

exists at least one path from u to v. Reachability information of different parts and entities in the network from each other, retrieve of such information after failures and disconnectivities in networks, and identification of the key nodes who play more pivotal roles in these reachabilities are some instances of reachability analysis which we address in Chapter 6.

Many networks are dynamically changing and prone to failures. However, it is inefficient to use the regular reachability methods in large and dense networks with high volume of reachability queries whenever a failure occurs in the network. We present a *dynamic reachability* method in the form of a pre-computed oracle which is cable of answering to reachability queries efficiently both in the case of having failures or no failure in a general directed network. Founded on the notion of reachability, we also extend the definition of *articulation points* to the directed networks which is originally defined for undirected networks, known as cut vertices as well. We also provide a formulation to compute the articulation points of a network and show that a similar formulation can quantify the load balancing over nodes of a network. Load balancing is important for network robustness against targeted attacks. Through extensive experiments, we evaluate the load balancing in several specific-shaped networks and real-world networks.

We also study the additional information associated with reachability such as how long (e.g., in terms of number of intermediate nodes to be traversed or some other measures of time or cost) or how many possible ways (e.g., in terms of paths) for node s to reach node t. Such information is essential for selecting paths for packet routing or information/commodity delivery, flow scheduling, power management, traffic control, load balancing and so forth in communication and computer networks, power grids and transportation networks. We propose a *pivotaility* metric which characterizes how pivotal a role that a node k or a subset of nodes S may play in the reachability from source node s to target node t in a given network. Through simulated and real-world network examples, we demonstrate that our metric provides a powerful ranking tool for the nodes based on their pivotality in the reachability.

0.4 Routing

After determining which entities are reachable from other entities, the next question is how to route a flow via a sequence of links and medial entities. Routing is the process of selecting a path for a flow in a network and is considered as one of the most essential decision making tasks in different types of networks, from computer networks to transportation networks. We study the routing problem under three different subjects: 1- Routing continuum as a generative model to generate a continuum from shortest path to (random walk) all path by tuning a control parameter, 2- Distance oracle to answer single-source shortest path (SSSP) queries in large networks efficiently, 3- Distance sensitivity oracle to find the replacement paths after some failures in the network efficiently. Theses theories have been developed consecutively and each subject is founded up on the previous one.

0.4.1 Routing Continuum

Shortest path is the most well known routing scheme which is desirable in many applications. However, having alternative paths is beneficial in many cases such as congestion reduction in data networks, avoiding complete predictability of the routing strategy, and increasing the robustness of the network. Hence, there is a growing literature on proposing strategies to generate multiple paths and avoid solely relaying on the shortest path [17, 54, 91, 108].

In Chapter 7, we develop a routing continuum method which generates a continuum from shortest path to all path by tuning a parameter. Our method provides a closed form formulation for computing the distances and an efficient routing strategy at the same time, unlike the existing routing continuum methods that either propose a formulation for computing the distances but no routing strategy, or suggest a routing strategy to determine the edge usages but no efficient formulation for computing the distances, . In addition, it generalizes the routing for cases with multiple targets. The other advantage of our method is building a unifying framework for network measure computations such as centrality measures, distance measures, and topological index. This method also proposes a novel shortest path method with the same complexity of existing shortest path methods but with the advantage of additional features such as efficiently providing

a distance oracle to address SSSP and replacement path problems 0.4.20.4.3.

0.4.2 Distance Oracle

Single-source shortest path (SSSP) and all-pair shortest path (APSP) form two main types of shortest path problems in which the shortest path from one source node to all the other nodes and between all the pairs of nodes are computed, respectively. However, particular applications might require something in the middle: answering several SSSP queries but not APSP. In such cases, an algorithm with prepossessing time faster than APSP and query time faster than SSSP is obviously superior compared with the existing SSSP and APSP algorithms. In Chapter 8, we propose an oracle which is precomputed once and is capable of answering to SSSP queries very efficiently.

0.4.3 Distance Sensitivity Oracle

When a network is prone to failures, it is very expensive to compute the shortest paths every time from scratch. Distance sensitivity oracle provides a solution for finding the new shortest paths faster and with lower cost by one-time pre-computation of an oracle. However, almost all of the solutions presented in the literature are supporting only the case of single failure and devising an efficient oracle for the case of multiple-failure was still considered an open problem. In Chapter 9, we present a novel oracle and replacement path method which addresses the multiple-failure case with complexity equal or comparable to that of single-failure case.

0.5 Summary of Contributions

- We develop a theoretical framework founded on Markov model theories and introduce new Markov metrics whose effectiveness are demonstrated in different applications in the rest of the thesis [60, 59].
- We propose the Heat Conduction (HC) influence model to capture the influence cascades in social networks which has favorable real world interpretations, and unifies, generalizes, and extends the existing non-progressive models. We develop a scalable and provably near-optimal solution for influence maximization problem

under HC. We are the first to present a scalable solution for influence maximization under non-progressive LT model, as a special case of HC model. We demonstrate high performance and scalability of our algorithm via extensive experiments on large networks and present the first ever real non-progressive cascade dataset [56].

- We investigate potential strategies for mitigating failure cascade in inter-dependent multi-layered networks in terms of different coupling structures between the layers of network [62].

- We develop a dynamic reachability oracle to answer efficiently to reachability queries in failure-prone networks with no update required and $O(1)$ query time [60].

- We extend the notation of articulation point to directed network which also supports the definition of cut-vertex for undirected networks as a special case, and provide formulation to find the articulation points as well as load balancing of networks. We illustrate the load balancing across the nodes in several synthetic and real networks [60].

- We develop the pivotality metric for assessing pivotality of nodes in the reachability of a source node to a target node and demonstrate its superiority over the similar metrics. We also show the effectiveness of the metric in a few real network examples [61].

- We develop a novel generative model to generate a continuum from shortest-path to all-path routing by tuning a control parameter with additional nice properties such as generalizablity to multi-target routings and a unifying framework for computing the existing network measures [58, 59].

- We devise a distance oracle and a distance sensitivity oracle to efficiently answer to SSSP queries in networks with no failures and (multiple) failures respectively [59].

Part II

Markov Tensor Theory

We have developed *Markov Tensor Theory* as a platform of powerful theories and tools founded on Markov chain theory and random walk methods which are very effective in complex networks analysis and provide efficient solutions to the network problems. Markov Tensor Theory is a layered-structure initiated from *fundamental tensor* as a generalization of fundamental matrix in Markov chain theory [120], advanced to the next layer of *avoidance fundamental tensor* by adding the option of avoiding (or transiting) a set of nodes on the paths, and completed by the last layer of *avoidance fundamental tensor in evaporation paradigm* in which the random walk can be evaporated on its way with different choices of evaporation intensity and not reaching the target (Fig. 1). By adding each layer, we provide more flexibility and creativity in designing the Markov chain metrics to capture and model more advanced cases in network problems. Note that the fundamental tensor is a special case of avoidance fundamental tensor when the avoiding set $\mathcal{F}$ is empty, and avoidance fundamental tensor is a special case of avoidance fundamental tensor in evaporation paradigm when the probability of following the original transition probabilities is equal to 1 and the evaporation probability, i.e. $1-\alpha$, becomes 0 (Fig. 1). Markov Tensor Theory is also a basis for generating the other Markov chain metrics, including hitting time and absorption probability, in a simple and unified way. We establish the effectiveness of Markov Tensor Theory in finding the most influential people in a social network for influence maximization 4, devising an oracle to efficiently answer dynamic reachability queries 6, computing the articulation points of directed networks 6, proposing a pivotality metric to rank the importance of nodes in reachabilities 6, developing a generative model for a routing continuum from shortest path to (random walk) all path 7, and devising a distance oracle which answers to single-source shortest path (SSSP) queries 8 and finds replacement paths in multiple failures efficiently9.

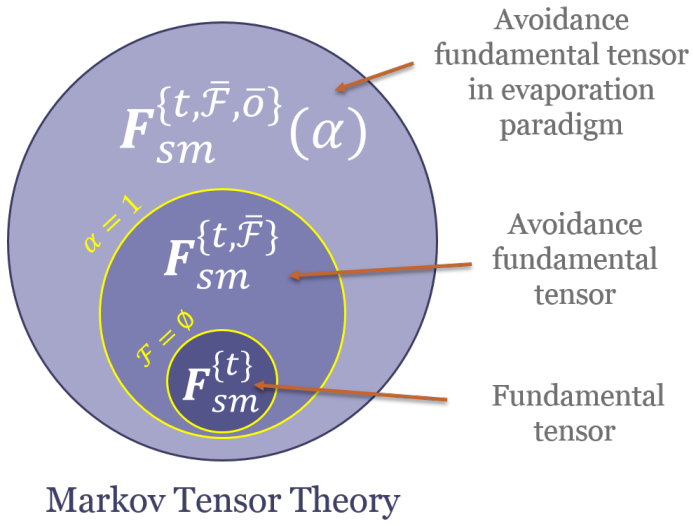


Figure 1: Layered structure of Markov Tensor Theory which has been proposed as an effective tool for complex network analysis in this dissertation.

In this Part of the dissertation, Chapter 1 overviews a preliminary on Markov chain theory, potential theory, and network theory, and provides a table of key notations used throughout this dissertation. Chapter 2 covers the developed theories on fundamental tensor and shows how the other well-known Markov chain classical metrics, such as hitting time and absorption probability, are related to fundamental tensor. This chapter also provides a complete library on Markov chain classical metrics theories and relations. In Chapter 3, we develop the avoidance fundamental tensor and a few other Markov chain avoidance metrics and present a reach set of theorems and relations on these metrics. We also introduce the evaporation paradigm in this chapter, but defer the complete definition of avoidance fundamental tensor in evaporation paradigm to Chapter 7 where the related application makes it easier to understand the metric. The effectiveness of Markov Tensor Theory in complex network analysis is elaborated in three forms of applications: 1) cascade, 2) reachability, and 3) routing, in the next three Parts of the dissertation.

Chapter 1

Preliminaries

1.1 Markov Chain

Discrete-time finite-space Markov chain is a sequence of random variables $(X_k)_{k>0}$ with values in countable and finite set V, i.e. $X : \Omega \to V$. Each i is called a state and V forms the state-space, $i \in V$. Random state X takes value i with probability $\mathbb{P}(X = i)$. The probability that in time k the Markov chain be in state j, i.e. $X_k = j$, given that it has been in state i at time $k-1$, i.e. $X_{k-1} = i$, is called the transition probability from state i to j and is denoted by P_{ij}, i.e. $\mathbb{P}(X_k = j | X_{k-1} = i) = P_{ij}$. According to Markov property, given the state of Markov chain at time $k-1$, X_k is independent of the states prior to time $k-1$: $\mathbb{P}(X_k = i_k | X_{k-1} = i_{k-1}, X_{k-2} = i_{k-2}, ..., X_0 = i_0) = \mathbb{P}(X_k = i_k | X_{k-1} = i_{k-1}) = P_{i_{k-1} i_k}$. Therefor, a Markov chain is fully described by its transition probability matrix $P = \{P_{ij}\}$. Note that Markov chains are mostly considered to be time-homogeneous, thus the transition probability matrix P is the same after each step (or transition). In other words, a finite Markov chain is a finite Markov process such that the transition probabilities do not depend on time k [72]. The m-step transition probability can be computed as the m-th power of the transition matrix P^m, e.g. $\mathbb{P}(X_{k+m} = j | X_k = i) = [P^m]_{ij}$ is the probability of being in state j after m steps starting from state i.

State j is reachable from state i if there exists some m that $[P^m]_{ij} \neq 0$. Also if this reachability is mutual, i and j fall in the same communication or equivalence class of states. Each Markov chain is composed of these smaller pieces, namely equivalence

classes, which yield a partial ordering of the chain indicating the possible directions of proceeding the process. The equivalence classes are referred as transient or recurrent sets. The former, once left, are never entered again. The latter, once entered, are never left again. If a chain is composed of only one equivalence class, it is called *irreducible*. In a irreducible Markov chain all the states are reachable from each other. Finite-state irreducible Markov chains have unique stationary distribution π. A Markov chain that is both irreducible and aperiodic is called ergodic and has a limiting distribution as well (which is its stationary distribution). For further details please refer to [72].

If a recurrent set is composed of only one state, it is called absorbing state. The corresponding row of an absorbing state in transition probability matrix is a vector of all 0's but its diagonal entry which is 1. A Markov chain is called absorbing if it has at least one absorbing state such that from each transient state at least one absorbing state is reachable. This condition requires that the states be either transient or absorbing, and there should be no recurrent class which contains more than one state. Assuming that states are ordered in the way that set of transient states $\mathcal{T}$ come first and set of absorbing states $\mathcal{A}$ come last, the transition matrix for an absorbing Markov chain takes the following block matrix form:

$$P = \begin{bmatrix} P_{\mathcal{T}\mathcal{T}} & P_{\mathcal{T}\mathcal{A}} \\ 0 & I_{\mathcal{A}\mathcal{A}} \end{bmatrix}, \tag{1.1}$$

where $I_{\mathcal{A}\mathcal{A}}$ is an identity matrix and P is row-stochastic.

The expected number of steps required that starting from state i the Markov chain hits state j for the first time is called (expected) hitting time, and the probability of ever hitting j is referred as the hitting probability. If j is absorbing this probability is called absorption probability. These two quantities can be computed from the *fundamental matrix* whose entries represent the expected number of visits of states. We refer to all of these quantities as *Markov chain classical metrics* which are discussed in more details in next chapter.

1.2 Potential Theory and Harmonic Functions

Potential theory is a mathematical framework shared with several physical theories such as Newton's theory of gravity, electrostatics, fluid flow, and the diffusion of heat. The

focus of this thesis is on potentials associated to a Markov chain and computing them for Markov chain models. Expected hitting time and absorption probability are examples of potentials associated to a Markov chain. Potentials, denoted by ϕ are the solutions of linear system equations and have the following general form [102]:

$$\phi_i = \mathbb{E}_i(\sum_{k<T} c(X_k) + f(X_T)1_{T<\infty}), \tag{1.2}$$

where $\mathbb{E}_i$ is the expectation of the processes started from state i, $1_{T<\infty}$ is an indicator function, and T is the stopping time (hitting time is an example of stopping time). The state-space V is divided into two disjoint sets of states: interior states $\mathcal{I}$ and boundary states $\mathcal{B}$. Functions c and f are the cost functions defined over interior set and boundary set respectively: $(c_i : i \in \mathcal{I})$ and $(f_i : i \in \mathcal{B})$. If these two functions are non-negative and T is the hitting time, potential ϕ can be found from the following relation [102]:

$$\begin{cases} \phi = P\phi + c & \text{in } \mathcal{I} \\ \phi = f & \text{in } \mathcal{B} \end{cases}, \tag{1.3}$$

where P is the transition probability matrix. c for the expected hitting time is a vector of all 1's, and for absorbing probabilities (known as hitting probabilities as well) is zero. When $c = 0$ and $\phi = P\phi$, potential function ϕ is called *harmonic* in $\mathcal{I}$. The harmonic function over interior set is fully determined from potential values over the boundary set:

$$\phi_i = \mathbb{E}_i(f(X_T)) = \sum_{j \in \mathcal{B}} f_j \mathbb{P}(X_T = j), \tag{1.4}$$

where $i \in \mathcal{I}$.

1.3 Networks

A network can be abstractly modeled as a *weighted* and *directed* graph, denoted by $G = (V, E, A)$. Here V is the set of nodes in the network such as routers or switches in a communication network or users in a social network; E is the set of (*directed*) edges representing the (physical or logical) connections between nodes (e.g., a communication link from a node i to a node j) or entity relations (e.g., follower-followee relation between two users). The *affinity* (or adjacency) matrix $A = [a_{ij}]$ is assumed to be nonnegative,

i.e., $a_{ij} \geq 0$, where $a_{ij} > 0$ if and only if $\langle i, j \rangle \in E$. We remark that here a_{ij} captures certain "affinity" or "closeness" from node i to node j.

A random walk over G is modeled by a Markov chain, where the nodes of G represent the states of the Markov chain. Walk is a path in G whose nodes/states can be repeated. The Markov chain is fully described by its transition probability matrix: $P = D^{-1}A$, where A is the adjacency matrix of network G, $D = diag[d_i]$, and $d_i = \sum_j a_{ij}$ (d_i is often referred to as the (out-)degree of node i and D is the (diagonal) degree matrix). Throughout the thesis, the words "node" and "state", and "network" and "Markov chain" are used interchangeably. In addition, the target nodes in G can be represented as absorbing states in the Markov chain as once being hit, the random walk stops walking around. If the network G is strongly connected, the associated Markov chain is irreducible and all the stationary probabilities π are strictly positive according to Perron-Frobenius theorem [55]. For undirected and connected G, the associated Markov chain is reversible and the stationary probabilities are a scalar multiple of node degrees: $\pi_i = \frac{d_i}{\sum_i d_i}$.

1.4 Table of Notations

In Table (1.1), the notations used throughout the thesis have been listed for an easier reference.

Table 1.1: Table of Notations

P_{sm}	sm-th entry of transition matrix P. = Transition probability from s to m.
$[P^k_{\mathcal{TT}}]_{sm}$	sm-th entry belonging to the k-th power of $P_{\mathcal{TT}}$. = Probability of being in node m in k-th step when starting from node s.
$[P_{\mathcal{TA}}]_{mt}$	mt-th entry of $P_{\mathcal{TA}}$, where $\mathcal{T}$ is the transient set and $\mathcal{A}$ is the absorbing set.
$\boldsymbol{F}$	Fundamental tensor.
$\boldsymbol{F}^{\{t\}}_{sm}$	smt-th entry of Fundamental tensor.
$F^{\{t\}}$	The t-th cross-section of fundamental tensor = Fundamental matrix for target node t.
$F^{\{t\}}_{sm}$	sm-th entry of Fundamental matrix $F^{\{t\}}$. = Expected number of times the random walk passes through m when it starts from s and before hitting t.
$F^{\{t,\overline{\mathcal{F}}\}}$	Avoidance fundamental matrix for target node t and avoiding set $\mathcal{F}$.
$F^{\{t,\overline{\mathcal{F}},\bar{o}\}}(\alpha)$	Avoidance fundamental matrix in evaporation paradigm for target node t and avoiding set $\mathcal{F}$.
$H^{\{t\}}_s$	Expected hitting time from s to t. = Expected number of steps required to hit t for the first time when starting from s.
$U^{\{t\}}_s$	Expected hitting cost from s to t. = Expected cost required to hit t for the first time when starting from s.
$Q^{\{t,\overline{\mathcal{F}}\}}_s$	Absorption probability from s to t avoiding set $\mathcal{F}$.
$\mathcal{A}$	General label for absorbing set.
$\mathcal{T}$	General label for transient set.
Z_{sm}	The set of all walks from s to m. In contrast to *paths*, the nodes can be repeated in *walks*.
$Z_{sm}(l)$	The set of all walks from s to m with total length of l.
$Z_{sm}(k,l)$	The set of all walks from s to m with k number of steps and total length of l. In unweighted networks $l = k$.
$\zeta_j \in Z_{sm}(k,l)$	The j-th walk from set $Z_{sm}(k,l)$. V_{ζ_j} is the set of nodes that ζ_j passes. = Walk ζ_j is specified by its sequence of nodes $v_0 v_1 ... v_k$, where $v_i \in V_{\zeta_j}, 0 \leq i \leq k$, and $v_0 = s$ and $v_k = m$.
$\Pr_{\zeta_j}$	Probability of walk ζ_j, $\Pr_{\zeta_j} = P_{sv_1} P_{v_1 v_2} ... P_{v_{k-1} m}$. $\sum_{\zeta_j \in Z_{sm}(k)} \Pr_{\zeta_j} = \begin{cases} [P^k_{\mathcal{TT}}]_{sm} & \text{if } m \in \mathcal{T} \\ [P^{k-1}_{\mathcal{TT}} P_{\mathcal{TA}}]_{sm} & \text{if } m \in \mathcal{A} \end{cases}$.
l_{ζ_j}	Length of walk ζ_j, $l_{\zeta_j} = \sum_{i=1}^{k} w_{v_{i-1} v_i}$.
k_{ζ_j}	Number of steps in walk ζ_j.
$Z^{\overline{\mathcal{F}}}_{sm}$	The set of all walks from s to m that avoid the nodes in set $\mathcal{F}$.

Chapter 2

Fundamental Tensor and Other Markov Chain Classical Metrics

2.1 Introduction

In this chapter, we introduce the *fundamental tensor* as a generalization of fundamental matrix and review some of Markov chain classical metrics and their different forms of definitions presented in literature. We show the connections of these metrics to fundamental tensor and how they all can be computed from the fundamental tensor in a unified way. At the end, we present some lemmas and relations for Markov chain classical metrics which would be useful for applications and following chapters.

2.2 Fundamental Matrix

Expected number of visits metric counts the expected number of passages or visits of a state in a Markov chain, given the starting (or source) state. In other words, it counts the expected number of times that the random walk passes through a state when it starts from a source state. This metric has an implicit stopping criteria for the counts which is visiting a target state (or a set of target states) for the first time. Visiting a state for the first time is called *hitting* that state as well. This metric depends on the source state, visiting (medial) state, and the target state. Fundamental matrix is composed of the expected number of visits for all pairs of source states and medial states, with the

same target state for all the pairs. Different forms of fundamental matrix (or expected number of visits) definition presented in literature [72, 102] are reviewed in the following.

Matrix form: The fundamental matrix of absorbing chain P is defined as follows:

$$F = (I - P_{\mathcal{TT}})^{-1}, \tag{2.1}$$

where entry F_{ij} represents the expected number of passages through state j, starting from state i, and before absorption by any of absorbing states [120]. Knowing that the k-th power of the transient-to-transient part of P, i.e. $P^k_{\mathcal{TT}}$, calculates the probability of being in different transient states in k-th step, and writing Eq. (2.1) as a geometric series, $(I - P_{\mathcal{TT}})^{-1} = I + P_{\mathcal{TT}} + P^2_{\mathcal{TT}} + \dots$, the concept of expected number of passages through the transient states would be clear to understand.

Stochastic form: The indicator function $1_{\{X_k=m\}}$ is the random variable equal to 1 if $X_k = m$ and 0 otherwise. The number of visits ν_m to m is written in terms of indicator functions as $\nu_m = \sum_{k=0}^{\infty} 1_{\{X_k=m\}}$. The expected number of times visiting m when the walk starts at s is denoted by [102]:

$$\begin{aligned} F_{sm} &= \mathbb{E}_s(\nu_m) = \mathbb{E}_s \sum_{k=0}^{\infty} 1_{\{X_k=m\}} = \sum_{k=0}^{\infty} \mathbb{E}_s(1_{\{X_k=m\}}) \\ &= \sum_{k=0}^{\infty} \mathbb{P}(X_k = m | X_0 = s) = \sum_{k=0}^{\infty} [P^k_{\mathcal{TT}}]_{sm}, \end{aligned} \tag{2.2}$$

The stopping criteria is hitting t for the first time, so this quantity is denoted by $F^{\{t\}}_{sm}$ more precisely. In an irreducible chain, by having such stopping criteria, k cannot be ∞ and $F^{\{t\}}_{sm} = \sum_{k<\infty} [P^k_{\mathcal{TT}}]_{sm}$ is finite. Note that in order to have finite value for $F^{\{t\}}_{sm}$, it is enough that t be reachable from all the node in network and the irreducibility of network is not necessary. We call the matrix of $F^{\{t\}}_{sm}$ values constructed for every pairs of $s, m \in \mathcal{T}$ as fundamental matrix for target t. It is easy to see that the fundamental matrix can be computed from the powers of $P_{\mathcal{TT}}$

$$F^{\{t\}} = \sum_{k<\infty} P^k_{\mathcal{TT}} = I + \sum_{k=1}^{<\infty} P^k_{\mathcal{TT}} = (I - P_{\mathcal{TT}})^{-1}, \tag{2.3}$$

which is the derivation in Eq. (2.1).

Recursive form: The expected number of visits of a state is the summation of the original position's contribution and each of the steps' contribution. The original position contributes 1 if and only if $s = m$ or $X_0 = m$ equivalently.

$$\begin{aligned} F_{sm} &= 1_{\{X_0=m\}} + \sum_{j \in \mathcal{T}} P_{sj} F_{jm} \\ \Rightarrow F &= I + P_{\mathcal{T}\mathcal{T}} F \\ \Rightarrow F &= (I - P_{\mathcal{T}\mathcal{T}})^{-1} \end{aligned} \tag{2.4}$$

2.2.1 Fundamental matrix for a Set of Targets

Let $T = \{t_1, ..., t_c\}$ be a set of targets. Fundamental matrix entry $F_{sm}^{\{T\}}$ for this set of targets represents the expected number of visits to m before hitting *either* of the states in T for the first time. It is a simple generalization of fundamental matrix for single target:

$$F^{\{T\}} = I + \sum_{k=1}^{<\infty} P_{\mathcal{T}\mathcal{T}}^k = (I - P_{\mathcal{T}\mathcal{T}})^{-1}, \tag{2.5}$$

where $\mathcal{T} = V \setminus T$.

2.2.2 Fundamental Tensor

Fundamental tensor $\boldsymbol{F}$ is a generalization of fundamental matrix by stacking up the fundamental matrices constructed for each node t as the target node in a strongly connected network. In other words, the fundamental tensor exists for any triplets of nodes (s, m, t) in a strongly connected network and the t-th cross section of fundamental tensor is computed from the corresponding fundamental matrix $F^{\{t\}}$:

$$\boldsymbol{F}_{smt} = \begin{cases} F_{sm}^{\{t\}} & \text{if } s, m \neq t \\ 0 & \text{if } s = t \text{ or } m = t \end{cases} \tag{2.6}$$

2.3 Hitting Time

The (expected) *hitting time*, also known in the literature as the *first transit time* or *first passage time*, has the concept of distance (or delay). This metric counts the expected

number of steps (or time steps) required to hit or visit the target state for the first time, so is called the expected hitting time (or hitting time in brief). This metric depends on both the source state that the random walk starts from and the target state which is the stopping criteria.

Matrix form: Expected absorption time, which is also known as (expected) hitting time or first passage time in literature, is calculated as follows:

$$\boldsymbol{h} = F\mathbf{1}, \tag{2.7}$$

where $\boldsymbol{h}$ is a column vector of length $|\mathcal{T}|$ representing the expected number of steps before absorption by either of the absorbing nodes and for each starting state.

Stochastic form: Let $G = (X_k)_{k>0}$ be a discrete-time Markov chain with transition matrix P. The hitting time of a node $t \in V$ is the random variable $\kappa_t : \Omega \to \{0, 1, 2, ...\} \cup \{\infty\}$ given by $\kappa_t = \inf\{\kappa \geq 0 : X_\kappa = t\}$, where we agree that the infimum of the empty set $\emptyset$ is ∞. The hitting time κ_t represents the number of steps that the walk takes until it hits t for the first time, and its expected value when the walk starts at s is denoted by [102]: $H_s^{\{t\}} = \mathbb{E}_s[\kappa_t] = \sum_{k<\infty} k\mathbb{P}(\kappa_t = k|X_0 = s) + \infty\mathbb{P}(\kappa_t = \infty|X_0 = s)$. Assuming that the target node t is reachable from all the other nodes in the network, case $\kappa_t = \infty$ does not occur. Note that in our cases, node t is visited only once and at the end of the walk so it can be considered as an absorbing node, where $\mathcal{A} = \{t\}$ and transient nodes are $\mathcal{T} = V \setminus \{t\}$. Source node s is usually considered to be any node other than t, i.e. $s \in \mathcal{T}$ and the number of steps to reach t is $k > 0$. Consequently, the following form is commonly used:

$$\begin{aligned} H_s^{\{t\}} &= \mathbb{E}_s[\kappa_t] = \sum_{k=1}^{<\infty} k\mathbb{P}(\kappa_t = k|X_0 = s) \\ &= \sum_{k=1}^{<\infty} k\mathbb{P}(X_k = t|X_0 = s) = \sum_{k=1}^{<\infty} k \sum_{m\in\mathcal{T}} [P_{\mathcal{T}\mathcal{T}}^{k-1}]_{sm}[P_{\mathcal{T}\mathcal{A}}]_{mt}, \end{aligned} \tag{2.8}$$

where $[P_{\mathcal{T}\mathcal{T}}^0]_{sm} = 1$ for $m = s$ and it is 0 otherwise.

Recursive form: The *recursive* relation of $H_s^{\{t\}}$, which is presented in many references, is proved in the following:

$$\begin{aligned}
H_s^{\{t\}} &= \sum_{k=1} k\mathbb{P}(X_k = t|X_0 = s) = \sum_{k=1} k \sum_{m\in\mathcal{N}_{out}(s)} \mathbb{P}(X_k = t, X_1 = m|X_0 = s) \\
&= \sum_{k=1} k \sum_{m\in\mathcal{N}_{out}(s)} \mathbb{P}(X_k = t|X_0 = s, X_1 = m)\mathbb{P}(X_1 = m|X_0 = s) \\
&= \sum_{k=1} k \sum_{m\in\mathcal{N}_{out}(s)} \mathbb{P}(X_k = t|X_0 = s, X_1 = m)P_{sm} \\
&= \sum_{m\in\mathcal{N}_{out}(s)} \sum_{k=1} (1 + (k-1))\mathbb{P}(X_k = t|X_0 = s, X_1 = m)P_{sm} \\
&= \sum_{m\in\mathcal{N}_{out}(s)} \sum_{k=1} (1 + (k-1))\mathbb{P}(X_{k-1} = t|X_0 = m)P_{sm} \qquad (2.9) \\
&= \sum_{m\in\mathcal{N}_{out}(s)} \sum_{k=1} \mathbb{P}(X_{k-1} = t|X_0 = m)P_{sm} + (k-1)\mathbb{P}(X_{k-1} = t|X_0 = m)P_{sm} \\
&= \sum_{m\in\mathcal{N}_{out}(s)} P_{sm} \sum_{k=1} \mathbb{P}(X_{k-1} = t|X_0 = m) + \sum_{m\in\mathcal{N}_{out}(s)} P_{sm} \sum_{k=1} (k-1)\mathbb{P}(X_{k-1} = t|X_0 = m) \\
&= \sum_{m\in\mathcal{N}_{out}(s)} P_{sm} + \sum_{m\in\mathcal{N}_{out}(s)} P_{sm} \sum_{k'=1} k'\mathbb{P}(X_{k'} = t|X_0 = m) \\
&= 1 + \sum_{m\in\mathcal{N}_{out}(s)} P_{sm} H_m^{\{t\}},
\end{aligned}$$

where (2.9) is due to two properties of Markov chains that networks are modeled by: 1) First order Markov chain and 2) Time-homogeneous Markov chain.

Relation between fundamental matrix and hitting time: The expected hitting time $H_s^{\{t\}}$ and the expected number of visits $F_{sm}^{\{t\}}$ are in close relation with each other

$$\begin{aligned}
H_s^{\{t\}} &= \sum_{k=1} k \sum_{m\in\mathcal{T}} [P_{\mathcal{TT}}^{k-1}]_{sm}[P_{\mathcal{TA}}]_{mt} = \sum_{k=1} k \sum_{m\in\mathcal{T}} [P_{\mathcal{TT}}^{k-1}]_{sm}(1 - \sum_{j\in\mathcal{T}} [P_{\mathcal{TT}}]_{mj}) \\
&= \sum_{k=1} k(\sum_{m\in\mathcal{T}} [P_{\mathcal{TT}}^{k-1}]_{sm} - \sum_{j\in\mathcal{T}} [P_{\mathcal{TT}}^{k}]_{sj}) = \sum_{m\in\mathcal{T}} \sum_{k=1} k([P_{\mathcal{TT}}^{k-1}]_{sm} - [P_{\mathcal{TT}}^{k}]_{sm}) \\
&= \sum_{m\in\mathcal{T}} [P_{\mathcal{TT}}^{k-1}]_{sm} = \sum_{m} F_{sm}^{\{t\}}, \qquad (2.10)
\end{aligned}$$

which is the same as the relation in (2.7).

2.3.1 Hitting Time for a Set of Targets

Let $T = \{t_1, ..., t_c\}$ be a set of targets. Hitting time for this set of targets is defined to be the expected number of steps to hit *either* of the states in T for the first time. In the following we define it mathematically.

The hitting time of a set $T \subset V$ is the random variable $\kappa_T : \Omega \to \{0, 1, 2, ...\} \cup \{\infty\}$ given by $\kappa_T = \inf\{\kappa \geq 0 : X_\kappa \in T\}$, where we agree that the infimum of the empty set $\emptyset$ is ∞. The expected value of κ_T when the walk starts at s is denoted by: $H_s^{\{T\}} = \mathbb{E}_s[\kappa_T] = \sum_{k<\infty} k\mathbb{P}(\kappa_T = k|X_0 = s) + \infty\mathbb{P}(\kappa_T = \infty|X_0 = s)$. Assuming that the target set T (at least one of its states) is reachable from all the other nodes in the network, case $\kappa_T = \infty$ does not occur. Since in our cases, set T is visited only once and at the end of the walk so it can be considered as an absorbing set, where $\mathcal{A} = T$ and transient nodes are $\mathcal{T} = V \setminus T$. Source node s is usually considered not to belong to set T, i.e. $s \in \mathcal{T}$ and the number of steps to reach T is $k > 0$. Consequently, the following form is commonly used:

$$
\begin{aligned}
H_s^{\{T\}} &= \mathbb{E}_s[\kappa_T] = \sum_{k=1}^{<\infty} k\mathbb{P}(\kappa_T = k|X_0 = s) && (2.11)\\
&= \sum_{i=1}^{c}\sum_{k=1}^{<\infty} k\mathbb{P}(X_k = t_i|X_0 = s) && (2.12)\\
&= \sum_{k=1}^{<\infty} k \sum_{m\in\mathcal{T}} [P_{\mathcal{T}\mathcal{T}}^{k-1}]_{sm} \sum_{i=1}^{c} [P_{\mathcal{T}\mathcal{A}}]_{mt_i}, && (2.13)
\end{aligned}
$$

where $[P_{\mathcal{T}\mathcal{T}}^0]_{sm} = 1$ for $m = s$ and it is 0 otherwise. Note that the second line of equalities is due to the fact that being in states $t_1, ..., t_c$ are mutually exclusive events and the union turns to a summation: $\mathbb{P}(\kappa_T = k|X_0 = s) = \cup_{t_i\in T}\mathbb{P}(X_k = t_i|X_0 = s) = \sum_{i=1}^{c}\mathbb{P}(X_k = t_i|X_0 = s)$.

Relation between fundamental matrix and hitting time for a set of targets: The expected hitting time $H_s^{\{T\}}$ and the expected number of visits $F_{sm}^{\{T\}}$ are in close

relation with each other

$$\begin{aligned} H_s^{\{T\}} &= \sum_{k=1} k \sum_{m\in\mathcal{T}} [P_{\mathcal{TT}}^{k-1}]_{sm} \sum_{i=1}^{c} [P_{\mathcal{TA}}]_{mt_i} = \sum_{k=1} k \sum_{m\in\mathcal{T}} [P_{\mathcal{TT}}^{k-1}]_{sm} (1 - \sum_{j\in\mathcal{T}} [P_{\mathcal{TT}}]_{mj}) \\ &= \sum_{k=1} k (\sum_{m\in\mathcal{T}} [P_{\mathcal{TT}}^{k-1}]_{sm} - \sum_{j\in\mathcal{T}} [P_{\mathcal{TT}}^{k}]_{sj}) = \sum_{m\in\mathcal{T}} \sum_{k=1} k ([P_{\mathcal{TT}}^{k-1}]_{sm} - [P_{\mathcal{TT}}^{k}]_{sm}) \\ &= \sum_{m\in\mathcal{T}} [P_{\mathcal{TT}}^{k-1}]_{sm} = \sum_{m} F_{sm}^{\{T\}}, \end{aligned} \tag{2.14}$$

2.3.2 Commute Time

Commute time is the summation of the hitting time from source to target and vice versa:

$$C_{st} = H_s^{\{t\}} + H_t^{\{s\}} \tag{2.15}$$

It is obvious that Commute time is a symmetric quantity while Hitting time is generally not, whether the underlying graph is directed or undirected.

2.4 Absorption Probability

The absorption probability matrix Q is defined as [120]:

$$Q = FP_{\mathcal{TA}}, \tag{2.16}$$

where its dimension is $|\mathcal{T}| \times |\mathcal{A}|$. The ij-th entry of Q is the probability of absorption by the absorbing state j when the chain starts from state i. The rows sum up to one, since starting from any state the chain (or random walker) will end up being absorbed by one of the absorbing states. The formulation above simply says that to obtain the probability of getting absorbed by a given absorbing state, we add up the probabilities of going there from all of the transient states, weighted by the number of times we expect to be in those transient states [120].

2.4.1 Normalized Fundamental Matrix

We define the normalized fundamental matrix as the fundamental matrix whose entries are normalized by corresponding diagonal entry:

$$\hat{F}_{sm}^{\{t\}} = \frac{F_{sm}^{\{t\}}}{F_{mm}^{\{t\}}} \tag{2.17}$$

In Theorem (2) we show that the sm-th entry of normalized fundamental matrix for target t is in fact the absorption probability by node m sooner than node t (starting from s).

2.5 Hitting Cost

The (expected) hitting cost, also known as average first-passage cost generalizes the (expected) hitting time by assigning a cost to each transition. Hitting cost from s to t denoted by $U_s^{\{t\}}$ is the average cost incurred by the random walker starting from state s to hit state t for the first time. The cost of edge e_{ij} transition is given by w_{ij}. Notice that hitting time $H_s^{\{t\}}$ is a special case of hitting cost $U_s^{\{t\}}$ obtained when $w_{ij} = 1$ for all e_{ij} edges. The hitting cost was first introduced by Fouss et al. [50] along with an intuitive recusrsive formulation for it. In this part, we first provide a fundamental and rigorous definition for hitting cost in stochastic form and then derive the recursive form as well as the matrix form (which is a closed form formulation useful for applications) from this definition.

Stochastic form: Let $G = (X_k)_{k>0}$ be a discrete-time Markov chain with transition matrix P and weight matrix W. The hitting cost of a node $t \in V$ is a random variable $\eta_t : \Omega \to \mathcal{C}$ given by $\eta_t = \inf \{\eta \geq 0 : \exists k, X_k = t, \sum_{i=1}^{k} w_{X_{i-1}X_i} = \eta\}$. $\mathcal{C}$ is a countable set and we agree that the infimum of the empty set $\emptyset$ is ∞. The hitting cost η_t represents the total length of steps that the walk takes until it hits t for the first time, and its

expected value when the walk starts at s is denoted by:

$$\begin{aligned}
U_s^{\{t\}} &= \mathbb{E}_s[\eta_t] = \sum_{l\in\mathcal{C}} l\mathbb{P}(\eta_t = l|X_0 = s) \\
&= \sum_{l\in\mathcal{C}} l \sum_{k=1}^{<\infty} \mathbb{P}(X_k = t, \sum_{i=1}^{k} w_{X_{i-1}X_i} = l|X_0 = s) \\
&= \sum_{l\in\mathcal{C}} l \sum_{k=1}^{<\infty} \sum_{\substack{v_{k-1}\in \\ \mathcal{N}_{in}(t)\cap\mathcal{N}_{out}(v_{k-2})}} \cdots \sum_{\substack{v_1\in \\ \mathcal{N}_{in}(v_2)\cap\mathcal{N}_{out}(s)}} \\
&\quad \mathbb{P}(X_k = t, X_{k-1} = v_{k-1}, ..., X_2 = v_2, X_1 = v_1, \sum_{i=1}^{k} w_{X_{i-1}X_i} = l|X_0 = s) \\
&= \sum_{l\in\mathcal{C}} l \sum_{k=1}^{<\infty} \sum_{\zeta_j\in Z_{st}(k,l)} \Pr_{\zeta_j} \qquad (2.18) \\
&= \sum_{l\in\mathcal{C}} l \sum_{\zeta_j\in Z_{st}(l)} \Pr_{\zeta_j} \qquad (2.19) \\
&= \sum_{l\in\mathcal{C}} l\Pr_l, \qquad (2.20)
\end{aligned}$$

where $\Pr_l$ is the probability of hitting t in total length of l when starting from s, and is obtained from the aggregation of walk probabilities with length l. Therefor, the following three quantities are all the same: $\Pr_l = \sum_{\zeta_j\in Z_{st}(l)} \Pr_{\zeta_j} = \mathbb{P}(\eta_t = l|X_0 = s)$.

We can also continue (2.18) as follows to achieve another form of hitting cost:

$$\begin{aligned}
U_s^{\{t\}} &= \sum_{l\in\mathcal{C}} l \sum_{k=1}^{<\infty} \sum_{\zeta_j\in Z_{st}(k,l)} \Pr_{\zeta_j} \qquad (2.21) \\
&= \sum_{l\in\mathcal{C}} \sum_{k=1}^{<\infty} \sum_{\zeta_j\in Z_{st}(k,l)} l_{\zeta_j}\Pr_{\zeta_j} \\
&= \sum_{\zeta_j\in Z_{st}} l_{\zeta_j}\Pr_{\zeta_j} \qquad (2.22)
\end{aligned}$$

Matrix form: Hitting cost can be computed from the following *closed form* formulation:

$$
\begin{aligned}
U_s^{\{t\}} &= \sum_{\zeta_j \in Z_{st}} l_{\zeta_j} \Pr_{\zeta_j} = \sum_{\zeta_j \in Z_{st}} \Pr_{\zeta_j} \sum_{k=1}^{k_{\zeta_j}} w_{v_{k-1}v_k} && (2.23)\\
&= \sum_{\zeta_j \in Z_{st}} \sum_{k=1}^{k_{\zeta_j}} [\prod_{i=1}^{k} P_{v_{i-1}v_i}.(P_{v_k v_{k+1}} w_{v_k v_{k+1}}). \prod_{i=k+2}^{k_{\zeta_j}} P_{v_{i-1}v_i}] && (2.24)\\
&= \sum_{e_{xy} \in E} P_{xy} w_{xy} (\sum_{\zeta_j \in Z_{sx}} \Pr_{\zeta_j}).(\sum_{\zeta_i \in Z_{yt}} \Pr_{\zeta_i}) && (2.25)\\
&= \sum_{e_{xy} \in E} P_{xy} w_{xy} (\sum_{k} \sum_{\zeta_j \in Z_{sx}(k)} \Pr_{\zeta_j}).(\sum_{k} \sum_{\zeta_i \in Z_{yt}(k)} \Pr_{\zeta_i}) && (2.26)\\
&= \sum_{e_{xy} \in E} P_{xy} w_{xy} (\sum_{k} [P_{\mathcal{T}\mathcal{T}}^{k}]_{sx}).(\sum_{k} [P_{\mathcal{T}\mathcal{T}}^{k-1} P_{\mathcal{T}\mathcal{A}}]_{yt}) && (2.27)\\
&= \sum_{e_{xy} \in E} P_{xy} w_{xy} F_{sx}^{\{t\}} Q_y^{\{t\}} && (2.28)\\
&= \sum_{e_{xy} \in E} P_{xy} w_{xy} F_{sx}^{\{t\}} && (2.29)\\
&= \sum_{x} F_{sx}^{\{t\}} \sum_{y \in \mathcal{N}_{out}(x)} P_{xy} w_{xy} && (2.30)\\
&= \sum_{x} F_{sx}^{\{t\}} r_x, && (2.31)
\end{aligned}
$$

where $r_x = \sum_{y \in \mathcal{N}_{out}(x)} P_{xy} w_{xy}$ is the average out-going cost of node x. In the equations above, (2.25) is concluded from multiplication principle, (2.27) can be found in Table (1.1), and (2.29) is due to the fact that $Q_y^{\{t\}} = 1$ when having t as the only absorbing node in the network (which is assumed to be reachable from all the other nodes in the network). Therefore, the following matrix form is obtained:

$$U^{\{t\}} = F^{\{t\}} \boldsymbol{r}, \tag{2.32}$$

where $\boldsymbol{r}$ is the vector of r_x's.

Recursive form: the recursive computation of $U_s^{\{t\}}$ which was suggested by Fouss

et al. [50] as well can be derived from the stochastic form of $U_s^{\{t\}}$:

$$\begin{aligned}
U_s^{\{t\}} &= \sum_{l\in\mathcal{C}} l\mathbb{P}(\eta_t = l|X_0 = s) = \sum_{l\in\mathcal{C}} l \sum_{m\in\mathcal{N}_{out}(s)} \mathbb{P}(\eta_t = l, X_1 = m|X_0 = s) \\
&= \sum_{l\in\mathcal{C}} l \sum_{m\in\mathcal{N}_{out}(s)} \mathbb{P}(\eta_t = l|X_0 = s, X_1 = m)\mathbb{P}(X_1 = m|X_0 = s) \\
&= \sum_{l\in\mathcal{C}} l \sum_{m\in\mathcal{N}_{out}(s)} \mathbb{P}(\eta_t = l|X_0 = s, X_1 = m)P_{sm} \\
&= \sum_{m\in\mathcal{N}_{out}(s)} \sum_{l\in\mathcal{C}} l\mathbb{P}(\eta_t = l|X_0 = s, X_1 = m)P_{sm} \\
&= \sum_{m\in\mathcal{N}_{out}(s)} \sum_{l\in\mathcal{C}} (w_{sm} + (l - w_{sm}))\mathbb{P}(\eta_t = l - w_{sm}|X_0 = m)P_{sm} \\
&= \sum_{m\in\mathcal{N}_{out}(s)} \sum_{l\in\mathcal{C}} w_{sm}\mathbb{P}(\eta_t = l - w_{sm}|X_0 = m)P_{sm} \\
&\quad +(l - w_{sm})\mathbb{P}(\eta_t = l - w_{sm}|X_0 = m)P_{sm} \\
&= \sum_{m\in\mathcal{N}_{out}(s)} w_{sm}P_{sm} \sum_{l\in\mathcal{C}} \mathbb{P}(\eta_t = l - w_{sm}|X_0 = m) \\
&\quad + \sum_{m\in\mathcal{N}_{out}(s)} P_{sm} \sum_{l\in\mathcal{C}} (l - w_{sm})\mathbb{P}(\eta_t = l - w_{sm}|X_0 = m) \\
&= \sum_{m\in\mathcal{N}_{out}(s)} w_{sm}P_{sm} + \sum_{m\in\mathcal{N}_{out}(s)} P_{sm} \sum_{l'\in\mathcal{C}} l'\mathbb{P}(\eta_t = l'|X_0 = m) \\
&= r_s + \sum_{m\in\mathcal{N}_{out}(s)} P_{sm}U_m^{\{t\}} \qquad (2.33)
\end{aligned}$$

2.5.1 Commute Cost

Commute cost Y_{st} is defined as the expected cost required to hit t for the first time and get back to s. Commute cost is a symmetric metric and is obtained from the following relation:

$$Y_{st} = U_s^{\{t\}} + U_t^{\{s\}} \qquad (2.34)$$

2.6 Relations, Lemmas, and Theorems

Lemma 1 ([18]). *Let* $\begin{bmatrix} L_{11} & l_{12} \\ l'_{21} & l_{nn} \end{bmatrix}$ *be an* $n\times n$ *irreducible matrix such that nullity(L)=1. Let* $M = L^+$ *be the pseudo-inverse of* L *partitioned similarly and* $(u', 1)L = 0$, $L(v; 1) =$

0, *where* u, v *are* $(n-1)$*-vectors, and* u' *and* v' *are their transposes respectively. Then the inverse of the* $(n-1)\times(n-1)$ *matrix* L_{11} *exists and is given by:*

$$L_{11}^{-1} = (I_{n-1} + vv')M_{11}(I_{n-1} + uu'), \tag{2.35}$$

where I_{n-1} *denotes the* $(n-1)\times(n-1)$ *identity matrix.*

Note that node n in the Lemma above can be substituted by any other node (index).

Lemma 2. *Fundamental matrix can be computed from the pseudo-inverse of Laplacian matrix* $L = \Pi(I-P)$*:*

$$F_{sm}^{\{t\}} = (L_{sm}^+ - L_{tm}^+ + L_{tt}^+ - L_{st}^+)\pi_m, \tag{2.36}$$

Proof. Substitute $L = \Pi(I-P)$ and $v = u = \mathbf{1}$ in Lemma (1). □

Corollary 1. *The entire fundamental tensor* $\boldsymbol{F}$ *of a strongly connected network can be computed in* $O(n^3)$ *complexity.*

Proof. The nullity of matrix $I-P$ for a strongly connected network is 1, so according to Eq. (2.36) all n^3 entries of fundamental tensor $\boldsymbol{F}$ can be computed from L^+ in constant time: $\boldsymbol{F}_{smt} = (L_{sm}^+ - L_{tm}^+ + L_{tt}^+ - L_{st}^+)\pi_m$. □

Corollary 2.

$$\sum_{st} F_{sm}^{\{t\}} = K\pi_m, \tag{2.37}$$

where K *is a constant independent of* m*.*

Proof.

$$\begin{aligned}
\sum_{st} F_{sm}^{\{t\}} &= \sum_{st}(L_{sm}^+ - L_{tm}^+ - L_{st}^+ + L_{tt}^+)\pi_m && (2.38)\\
&= 0 - 0 - 0 + (n\sum_t L_{tt}^+)\pi_m && (2.39)\\
&= K\pi_m, && (2.40)
\end{aligned}$$

where the second equality is proved due to the property that the column sum of $L^+ = (\Pi(I-P))^+$ is zero. K is in fact the Kirchhoff index which will be discussed in Sec. (7.7.4). □

Corollary 3. *Hitting time and commute time in terms of Laplacian matrix $L = \Pi(I-P)$ are as follows:*

$$H_i^{\{j\}} = \sum_m (L_{im}^+ - L_{jm}^+)\pi_m + L_{jj}^+ - L_{ij}^+, \tag{2.41}$$

$$C_{ij} = L_{ii}^+ + L_{jj}^+ - L_{ji}^+ - L_{ij}^+, \tag{2.42}$$

Proof. Use Eq. (2.7) and (2.36). □

Note that we can also write the metrics in terms of random walk Laplacian matrix $L^R = I - P$ by simply the following substitution: $L_{im}^+ - L_{ij}^+ = \frac{L_{im}^{R\,+}}{\pi_m} - \frac{L_{ij}^{R\,+}}{\pi_j}$.

Corollary 4. *Hitting cost and commute cost in terms of Laplacian matrix $L = \Pi(I-P)$ are as follows:*

$$U_i^{\{j\}} = \sum_m (L_{im}^+ - L_{jm}^+ + L_{jj}^+ - L_{ij}^+)b_m, \tag{2.43}$$

$$Y_{ij} = (L_{im}^+ - L_{jm}^+ + L_{jj}^+ - L_{ij}^+)\sum_m b_m, \tag{2.44}$$

where $b_m = r_m\pi_m$ and $r_m = \sum_{k\in\mathcal{N}_{out}(m)} P_{mk}w_{mk}$.

Proof. Use Eq. (2.32) and (2.36). □

From Eq. (2.42) and (2.44) it can be seen that commute cost is a multiple scalar of commute time.

Theorem 1 (Incremental Computation of Fundamental Matrix)**.** *The fundamental matrix for target set of $\mathcal{S}_1 \cup \mathcal{S}_2$ can be computed from the fundamental matrix for target set $\mathcal{S}_1$:*

$$F_{im}^{\{\mathcal{S}_1,\mathcal{S}_2\}} = F_{im}^{\{\mathcal{S}_1\}} - F_{i\mathcal{S}_2}^{\{\mathcal{S}_1\}}(F_{\mathcal{S}_2\mathcal{S}_2}^{\{\mathcal{S}_1\}})^{-1}F_{\mathcal{S}_2m}^{\{\mathcal{S}_1\}}, \tag{2.45}$$

where the subscripts represent the rows and columns selected from the matrix respectively, e.g. $F_{i\mathcal{S}_2}^{\{\mathcal{S}_1\}}$ denotes the row i and the columns corresponding to set $\mathcal{S}_2$ of the fundamental matrix $F^{\{\mathcal{S}_1\}}$.

Proof. Consider matrix $M = I - P_{\mathcal{T}\mathcal{T}}$ where the absorbing set is $\mathcal{A} = \mathcal{S}_1$ and the transient set $\mathcal{T} = V \setminus \mathcal{S}_1$. The inverse of M yields fundamental matrix $F^{\{\mathcal{S}_1\}}$, and the inverse of its sub-matrix obtained from removing rows and columns corresponding to

set $\mathcal{S}_2$ gives fundamental matrix $F^{\{\mathcal{S}_1,\mathcal{S}_2\}}$. The following equations show how to find a sub-matrix's inverse from the original matrix's inverse:

If A is invertible, we can factor matrix $M = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$ as follows

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} I & 0 \\ CA^{-1} & I \end{bmatrix} \begin{bmatrix} A & B \\ 0 & D - CA^{-1}B \end{bmatrix} \tag{2.46}$$

Inverting the both sides of the equation yields

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} A^{-1} & -A^{-1}BS^{-1} \\ 0 & S^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -CA^{-1} & I \end{bmatrix} \tag{2.47}$$

$$= \begin{bmatrix} A^{-1} + A^{-1}BS^{-1}CA^{-1} & -A^{-1}BS^{-1} \\ -S^{-1}CA^{-1} & S^{-1} \end{bmatrix} \tag{2.48}$$

$$= \begin{bmatrix} X & Y \\ Z & W \end{bmatrix}, \tag{2.49}$$

where $S = D - CA^{-1}B$. Therefor, A^{-1} can be computed from $A^{-1} = X - YW^{-1}Z$. □

Corollary 5. *The simplified form of (1) for single target is as follows:*

$$F_{im}^{\{j,k\}} = F_{im}^{\{j\}} - \frac{F_{ik}^{\{j\}} F_{km}^{\{j\}}}{F_{kk}^{\{j\}}} \tag{2.50}$$

Lemma 3.

$$P_{\mathcal{T}\mathcal{T}} F^{\{\mathcal{S}\}} = F^{\{\mathcal{S}\}} P_{\mathcal{T}\mathcal{T}} = F^{\{\mathcal{S}\}}, \tag{2.51}$$

where $\mathcal{T} \cup \mathcal{S} = \mathcal{V}$

Proof. It is a simple derivation from Eq. (2.1). □

Corollary 6 (The relation of fundamental matrix value at a node and its in-going neighbors)**.**

$$\begin{aligned} F_{im}^{\{j\}} &= \sum_{k \neq j} F_{ik}^{\{j\}} P_{km}, \qquad \text{if } i \neq m \\ F_{im}^{\{j\}} &= 1 + \sum_{k \neq j} F_{ik}^{\{j\}} P_{km}, \qquad \text{if } i = m \end{aligned} \tag{2.52}$$

Proof. It is a special case of Lemma (3). □

Theorem 2 (Absorption probability and normalized fundamental tensor)**.** *The absorption probability for absorbing set* $\{j\} \cup \mathcal{S}$ *can be found from the fundamental matrix for absorbing set* $\mathcal{S}$*:*

$$Q_i^{\{j,\overline{\mathcal{S}}\}} = \frac{F_{ij}^{\{\mathcal{S}\}}}{F_{jj}^{\{\mathcal{S}\}}} \tag{2.53}$$

Proof.

$$\begin{aligned} Q_x^{\{y,\overline{\mathcal{S}}\}} &= u_x' F^{\{y,\mathcal{S}\}} P_{\mathcal{T}\mathcal{A}} u_y \\ &= \sum_i F_{xi}^{\{y,\mathcal{S}\}} P_{iy} \\ &= \sum_i (F_{xi}^{\{\mathcal{S}\}} - \frac{F_{xy}^{\{\mathcal{S}\}} F_{yi}^{\{\mathcal{S}\}}}{F_{yy}^{\{\mathcal{S}\}}}) P_{i,} \\ &= \sum_i F_{xi}^{\{\mathcal{S}\}} P_{iy} - \frac{F_{xy}^{\{\mathcal{S}\}}}{F_{yy}^{\{\mathcal{S}\}}} \sum_i F_{yi}^{\{\mathcal{S}\}} P_{iy} \\ &= F_{xy}^{\{\mathcal{S}\}} - \frac{F_{xy}^{\{\mathcal{S}\}}}{F_{yy}^{\{\mathcal{S}\}}} (F_{yy}^{\{\mathcal{S}\}} - 1) \\ &= \frac{F_{xy}^{\{\mathcal{S}\}}}{F_{yy}^{\{\mathcal{S}\}}}, \end{aligned}$$

where the third equation and the fifth equation are direct results of Theorem (1) and Corollary (2.52) respectively. □

Relation 1 (Complementary relation of absorption probabilities)**.**

$$Q_i^{\{j,\overline{k}\}} = 1 - Q_i^{\{k,\overline{j}\}} \tag{2.54}$$

Proof. Based on the definition of Q, when having two absorbing nodes, the random walk eventually ends up in either of them. □

Relation 2 (Four relations between fundamental matrix and commute time)**.**

$$(1) \quad F_{ii}^{\{j\}} = \pi_i C_{ij} \tag{2.55}$$

$$(2) \quad \frac{F_{im}^{\{j\}}}{\pi_m} + \frac{F_{mi}^{\{j\}}}{\pi_i} = C_{ij} + C_{jm} - C_{im} \tag{2.56}$$

$$(3) \quad \frac{F_{im}^{\{j\}}}{\pi_m} + \frac{F_{ij}^{\{m\}}}{\pi_j} = C_{jm} \tag{2.57}$$

$$(4) \quad F_{im}^{\{j\}} + F_{jm}^{\{i\}} = \pi_m C_{ij} \tag{2.58}$$

Proof. Use (2.36) and (2.42). □

Relation 3 (The hitting time detour overhead in terms of other metrics)**.**

$$(1) \quad H_i^{\{j\}} + H_j^{\{m\}} - H_i^{\{m\}} = \frac{F_{im}^{\{j\}}}{\pi_m} \tag{2.59}$$

$$(2) \quad H_i^{\{j\}} + H_j^{\{m\}} - H_i^{\{m\}} = Q_i^{\{m,\bar{j}\}} C_{mj} \tag{2.60}$$

Proof. For the first equation use (2.36) and (2.41), and for the second one use the previous equation along with (2) and (2.55). □

Relation 4 (The hitting time for two target nodes in terms of hitting time for single target)**.**

$$H_i^{\{j,k\}} = H_i^{\{k\}} - Q_i^{\{j,\bar{k}\}} H_j^{\{k\}} = H_i^{\{j\}} - Q_i^{\{k,\bar{j}\}} H_k^{\{j\}}, \tag{2.61}$$

reforming the terms, we have: $H_i^{\{j\}} = H_i^{\{j,k\}} + Q_i^{\{k,\bar{j}\}} H_k^{\{j\}}$.

Proof. Sum two sides of equation (1) over m and substitute (2) in it. □

Relation 5 (Three inequalities for hitting time)**.**

$$(1) \quad H_i^{\{m\}} + H_m^{\{j\}} \geq H_i^{\{j\}} \qquad \textbf{\textit{(triangular inequality)}} \tag{2.62}$$

$$(2) \quad H_i^{\{j\}} \geq H_i^{\{j,m\}} \tag{2.63}$$

$$(3) \quad H_i^{\{m\}} + H_m^{\{j,k\}} \geq H_i^{\{j,k\}} \tag{2.64}$$

Proof. For the first equation use (2.41) and (2.68). For the second one use the aggregated form of (1) over m and the fact that F's entries are always non-negative. (I have not proved the third equation analytically, but based on (2.62) it seems to be correct!) □

Relation 6 (Two inequalities for fundamental matrix)**.**

$$(1) \quad F_{im}^{\{j\}} F_{kk}^{\{j\}} \geq F_{ik}^{\{j\}} F_{km}^{\{j\}} \tag{2.65}$$

$$(2) \quad F_{kk}^{\{j\}} \geq F_{ik}^{\{j\}} \tag{2.66}$$

Proof. For the first inequality use (1) and the fact that F's entries are always non-negative. For the second one use (2.55), (2.59), and (2.62). □

Relation 7 (Inequality for absorption probability)**.**

$$Q_i^{\{m,\bar{j}\}} \geq Q_i^{\{k,\bar{j}\}} Q_k^{\{m,\bar{j}\}} \tag{2.67}$$

Proof. use (2) and (2.65). □

Relation 8 (Inequality for Laplacian matrix)**.**

$$L_{im}^{+} + L_{kk}^{+} \geq L_{ik}^{+} + L_{km}^{+} \tag{2.68}$$

Proof. use (2.36) and the fact that F's entries are always non-negative. □

Relation 9 (Three relations for undirected networks (reversible Markov chain))**.**

$$(1) \quad \frac{F_{im}^{\{S\}}}{\pi_m} = \frac{F_{mi}^{\{S\}}}{\pi_i} \tag{2.69}$$

$$(2) \quad Q_i^{\{m,\bar{j}\}} C_m^{\{j\}} = Q_m^{\{i,\bar{j}\}} C_i^{\{j\}} \tag{2.70}$$

$$(3) \quad H_i^{\{m\}} + H_m^{\{j\}} + H_j^{\{i\}} = H_m^{\{i\}} + H_j^{\{m\}} + H_i^{\{j\}} \tag{2.71}$$

Proof. For the first equation use (2.36) and the fact that L^{+} is symmetric in undirected case. For the second one use (2), (2.36), (2.42) and the fact that L^{+} is symmetric in undirected case. The third one is proved by using (2.41) and the fact that L^{+} is symmetric in undirected case. □

Chapter 3

Avoidance Fundamental Tensor and Other Markov Chain Avoidance Metrics

3.1 Introduction

The existing theory on classical random walk metrics, including fundamental matrix, hitting time, hitting cost, and hitting (absorption) probability, is the result of imposing only the stopping criteria on the random walk, which is hitting the target node for the first time, and has no control or conditions on the visiting nodes in the middle of the walk. In this paper, we introduce the "avoidance" and "transit" random walk metrics which provide more flexibility in the design of random walk and impose new conditions on the walk to *avoid* or *transit* a specific node (or a set of nodes) before the stopping criteria. In particular, we introduce avoidance fundamental matrix, avoidance hitting time, transit hitting time, and avoidance hitting cost and establish theories which show the relation of the introduced metrics with each other.

3.2 Avoidance Hitting Time

We introduce the avoidance (expected) hitting time as the conditional expectation of hitting time conditioned on avoiding a subset of nodes. Recall that the hitting time

of a node $t \in V$ is the random variable $\kappa_t : \Omega \to \{0, 1, 2, ...\} \cup \{\infty\}$ given by $\kappa_t = \inf\{\kappa \geq 0 : X_\kappa = t\}$, where we agree that the infimum of the empty set $\emptyset$ is ∞. The hitting time κ_t represents the number of steps that the walk takes until it hits t for the first time. The avoidance (expected) hitting time from s to t conditioned on avoiding $\mathcal{F}$ is defined as follows

$$
\begin{aligned}
H_s^{\{t,\overline{\mathcal{F}}\}} = \mathbb{E}_s[\kappa_t | X_{i\leq\kappa_t} \notin \mathcal{F}] &= \sum_{k=1} k\mathbb{P}(X_k = t | X_{i\leq\kappa_t} \notin \mathcal{F}, X_0 = s) && (3.1)\\
&= \sum_{k=1} k\frac{\mathbb{P}(X_k = t, X_{i\leq k} \notin \mathcal{F} | X_0 = s)}{\mathbb{P}(X_{i\leq\kappa_t} \notin \mathcal{F} | X_0 = s)} && (3.2)\\
&= \frac{\sum_{k=1} k\mathbb{P}(X_k = t, X_{i\leq k} \notin \mathcal{F} | X_0 = s)}{\mathbb{P}(X_{i\leq\kappa_t} \notin \mathcal{F} | X_0 = s)} && (3.3)\\
&= \frac{\sum_{k=1} k\mathbb{P}(X_k = t, X_{i\leq k} \notin \mathcal{F} | X_0 = s)}{\sum_{k=1} \mathbb{P}(\kappa_t = k, X_{i\leq k} \notin \mathcal{F} | X_0 = s)} && (3.4)\\
&= \frac{\sum_{k=1} k\mathbb{P}(X_k = t, X_{i\leq k} \notin \mathcal{F} | X_0 = s)}{\sum_{k=1} \mathbb{P}(X_k = t, X_{i\leq k} \notin \mathcal{F} | X_0 = s)} && (3.5)\\
&= \frac{\sum_{k=1} k[P_{\mathcal{T}\mathcal{T}}^{k-1} P_{\mathcal{T}\mathcal{A}}]_{st}}{\sum_{k=1} [P_{\mathcal{T}\mathcal{T}}^{k-1} P_{\mathcal{T}\mathcal{A}}]_{st}}, && (3.6)
\end{aligned}
$$

where $\mathcal{T}$ here is $\mathcal{T} = V \setminus (\mathcal{F} \cup \{t\})$.

3.3 Avoidance Fundamental Tensor

We introduce the avoidance fundamental tensor as the conditional expectation of number of visits from an state while avoiding a subset of nodes $\mathcal{F}$. The indicator function $1_{\{X_k=m\}}$ is the random variable equal to 1 if $X_k = m$ and 0 otherwise. The number of visits ν_m to m is written in terms of indicator functions as $\nu_m = \sum_{k=0}^{\infty} 1_{\{X_k=m\}}$. Recall

that κ_t is the stopping criteria for the walk.

$$
\begin{aligned}
F_{sm}^{\{t,\overline{\mathcal{F}}\}} &= \mathbb{E}_s(\nu_m | X_{i\leq\kappa_t} \notin \mathcal{F}) = \sum_{k=0} \mathbb{E}_s(1_{\{X_k=m\}} | X_{i\leq\kappa_t} \notin \mathcal{F}) && (3.7)\\
&= \sum_{k=0} \mathbb{P}(X_k = m | X_{i\leq\kappa_t} \notin \mathcal{F}, X_0 = s) && (3.8)\\
&= \frac{\sum_{k=0} \mathbb{P}(X_k = m, X_{i\leq\kappa_t} \notin \mathcal{F} | X_0 = s)}{\mathbb{P}(X_{i\leq\kappa_t} \notin \mathcal{F} | X_0 = s)} && (3.9)\\
&= \frac{\sum_{k=0} \mathbb{P}(X_k = m, X_{i<k} \notin \mathcal{F}, X_{k<i\leq\kappa_t} \notin \mathcal{F} | X_0 = s)}{\mathbb{P}(X_{i\leq\kappa_t} \notin \mathcal{F} | X_0 = s)} && (3.10)\\
&= \frac{\sum_{k=0} \mathbb{P}(X_k = m, X_{i<k} \notin \mathcal{F} | X_0 = s) \mathbb{P}(X_k = m, X_{k<i\leq\kappa_t} \notin \mathcal{F} | X_0 = s)}{\mathbb{P}(X_{i\leq\kappa_t} \notin \mathcal{F} | X_0 = s)} &&\\
&= \frac{\sum_{k=0} \mathbb{P}(X_k = m, X_{i<k} \notin \mathcal{F} | X_0 = s) \mathbb{P}(X_{0<i\leq\kappa_t} \notin \mathcal{F} | X_0 = m)}{\mathbb{P}(X_{i\leq\kappa_t} \notin \mathcal{F} | X_0 = s)} && (3.11)\\
&= \frac{\sum_{k=0} [P_{\mathcal{T}\mathcal{T}}^k]_{sm} \sum_{k=1} [P_{\mathcal{T}\mathcal{T}}^{k-1} P_{\mathcal{T}\mathcal{A}}]_{mt}}{\sum_{k=1} [P_{\mathcal{T}\mathcal{T}}^{k-1} P_{\mathcal{T}\mathcal{A}}]_{st}} && (3.12)\\
&= \frac{F_{sm}^{\{t,\mathcal{F}\}} \sum_{k=1} [P_{\mathcal{T}\mathcal{T}}^{k-1} P_{\mathcal{T}\mathcal{A}}]_{mt}}{\sum_{k=1} [P_{\mathcal{T}\mathcal{T}}^{k-1} P_{\mathcal{T}\mathcal{A}}]_{st}}, && (3.13)
\end{aligned}
$$

Note that we use terms avoidance fundamental tensor and avoidance fundamental matrix interchangeably throughout the dissertation since we usually deal with a cross section of the tensor at a time.

3.4 Transit Hitting Time

Closely related to the avoidance hitting time is the notion of *transit* hitting time. For any third node k, the *transit* hitting time $H_s^{\{t,\check{k}\}}$ is the expected number of steps taken by a random walk which starts at node s and always traverse node k before hitting target node t for the first time. Using the avoidance hitting time, we can express $H_s^{\{t,\check{k}\}}$ as follows:

$$H_s^{\{t,\check{k}\}} = H_s^{\{k,\bar{t}\}} + H_k^{\{t\}}. \tag{3.14}$$

Using the avoidance and transit hitting times, we can now divide the paths (or walks) between a source node s to a target node t into two groups with respect to an arbitrary third node k: those paths that exclude or *avoid* node k and those that include or *transit* node k. The probability that a random walk takes either a path/walk from the first group vs. that from the second group is given by $Q_s^{\{t,\bar{k}\}}$ and $Q_s^{\{k,\bar{t}\}}$.

3.5 Avoidance Hitting Cost

We introduce the avoidance hitting cost as the conditional expectation of the hitting cost from source to target conditioned on avoiding a subset of nodes. In the following, we define it more rigorously in the stochastic form.

Let $G = (X_k)_{k>0}$ be a discrete-time Markov chain with transition matrix P and weight matrix W. Recall that the hitting cost of a node $t \in V$ is a random variable $\eta_t : \Omega \to \mathcal{C}$ given by $\eta_t = \inf\{\eta \geq 0 : \exists k, X_k = t, \sum_{i=1}^{k} w_{X_{i-1}X_i} = \eta\}$. Avoidance (expected) hitting cost from s to t conditioned on avoiding $\mathcal{F}$ is defined as follows:

$$
\begin{aligned}
U_s^{\{t,\overline{\mathcal{F}}\}} &= \mathbb{E}_s[\eta_t | X_k = t, X_{i\leq k} \notin \mathcal{F}] = \sum_{l\in\mathcal{C}} l\mathbb{P}(\eta_t = l | X_k = t, X_{i\leq k} \notin \mathcal{F}, X_0 = s) \\
&= \frac{\sum_{l\in\mathcal{C}} l\mathbb{P}(\eta_t = l, X_k = t, X_{i\leq k} \notin \mathcal{F} | X_0 = s)}{\mathbb{P}(X_k = t, X_{i\leq k} \notin \mathcal{F} | X_0 = s)} \qquad (3.15) \\
&= \frac{\sum_{l\in\mathcal{C}} l \sum_{k=1}^{<\infty} \mathbb{P}(\sum_{i=1}^{k} w_{X_{i-1}X_i} = l, X_k = t, X_{i\leq k} \notin \mathcal{F} | X_0 = s)}{\sum_{k=1}^{<\infty} \mathbb{P}(X_k = t, X_{i\leq k} \notin \mathcal{F} | X_0 = s)} \\
&= \frac{\sum_{l\in\mathcal{C}} l \sum_{k=1}^{<\infty} \sum_{\zeta_j \in Z_{st}^{\overline{\mathcal{F}}}(k,l)} \Pr_{\zeta_j}}{\sum_{k=1}^{<\infty} \sum_{\zeta_j \in Z_{st}^{\overline{\mathcal{F}}}(k)} \Pr_{\zeta_j}} \qquad (3.16) \\
&= \frac{\sum_{l\in\mathcal{C}} l \sum_{\zeta_j \in Z_{st}^{\overline{\mathcal{F}}}(l)} \Pr_{\zeta_j}}{\sum_{\zeta_j \in Z_{st}^{\overline{\mathcal{F}}}} \Pr_{\zeta_j}} \\
&= \frac{\sum_{l\in\mathcal{C}} l \sum_{\zeta_j \in Z_{st}^{\overline{\mathcal{F}}}(l)} \Pr_{\zeta_j}}{\sum_{l\in\mathcal{C}} \sum_{\zeta_j \in Z_{st}^{\overline{\mathcal{F}}}(l)} \Pr_{\zeta_j}} \\
&= \frac{\sum_{l\in\mathcal{C}} l \Pr_l^{\overline{\mathcal{F}}}}{\sum_{l\in\mathcal{C}} \Pr_l^{\overline{\mathcal{F}}}} \qquad (3.17)
\end{aligned}
$$

where $\Pr_l^{\{\overline{\mathcal{F}}\}}$ is the probability of hitting t in total length of l when starting from s and avoiding set $\mathcal{F}$. It is obtained from the aggregation of walk probabilities with length l which avoid set $\mathcal{F}$. Therefor, the following three quantities are all the same: $\Pr_l^{\{\overline{\mathcal{F}}\}} = \sum_{\zeta_j \in Z_{st}^{\overline{\mathcal{F}}}(l)} \Pr_{\zeta_j} = \mathbb{P}(\eta_t = l | X_{i\leq k} \notin \mathcal{F}, X_k = t, X_0 = s)$.

We can also continue (3.16) as follows to achieve another form of avoidance hitting

cost:

$$U_s^{\{t,\overline{\mathcal{F}}\}} = \frac{\sum_{l\in\mathcal{C}} l \sum_{k=1}^{<\infty} \sum_{\zeta_j \in Z_{st}^{\overline{\mathcal{F}}}(k,l)} \Pr_{\zeta_j}}{\sum_{k=1}^{<\infty} \sum_{\zeta_j \in Z_{st}^{\overline{\mathcal{F}}}(k)} \Pr_{\zeta_j}} \tag{3.18}$$

$$= \frac{\sum_{l\in\mathcal{C}} \sum_{k=1}^{<\infty} \sum_{\zeta_j \in Z_{st}^{\overline{\mathcal{F}}}(k,l)} l_{\zeta_j} \Pr_{\zeta_j}}{\sum_{k=1}^{<\infty} \sum_{\zeta_j \in Z_{st}^{\overline{\mathcal{F}}}(k)} \Pr_{\zeta_j}}$$

$$= \frac{\sum_{\zeta_j \in Z_{st}^{\overline{\mathcal{F}}}} l_{\zeta_j} \Pr_{\zeta_j}}{\sum_{\zeta_j \in Z_{st}^{\overline{\mathcal{F}}}} \Pr_{\zeta_j}} \tag{3.19}$$

3.6 Avoiding an External Node

The avoiding node can be an external (imaginary) node which is added to the network to model some phenomena in the network. Then the Markov chain avoidance metrics introduced in this chapter can be applied to avoid the external node instead of the real nodes. In the following, we present a few applications of adding an external state:

- **The exterior effect:** If the nodes in a network do not present all the sources of cascades or influences in the network, an external node (or state) can be added to model *exterior* effects. Then if the external node is avoided, the *interior* influence of network nodes is evaluated exclusively. Chapter (4) presents the cascade in social networks, where the activeness of nodes is defined to be adopting a product: a node is called active if she adopts product x and inactive otherwise. The activeness of nodes in a social network is a function of activeness of her friends (neighbors) in the network plus some exterior (or non-social) effect. We model the non-social (exterior) effect by an external node and compute the social (interior) effects by avoiding the external node.

- **Evaporation paradigm:** An evaporation paradigm corresponding to a network is formed by multiplying an evaporation factor $\alpha^{w_{ij}}$ into the transition probability of edge e_{ij}, i.e. $P_{ij}(\alpha) = \alpha^{w_{ij}} P_{ij}$ for all the edges in the network, where w_{ij} is the weight of the edge. To make the out-going probability of each node equal to 1 and the new transition probability matrix row-stochastic, one external node, called evaporation node o, is added to the network to which every node i is connected by a transition probability of $P_{io}(\alpha) = 1 - \sum_{j\in\mathcal{N}_{out}(i)} P_{ij}(\alpha) = 1 - \sum_{j\in\mathcal{N}_{out}(i)} \alpha^{w_{ij}} P_{ij}$. In

Chapter (7), we show that the avoidance hitting cost on the evaporation paradigm, where the evaporation node is avoided, yields a continuum from the shortest path distance to all-path distance when α ranges from 0 to 1.

3.7 Relations, Lemmas, and Theorems

Theorem 3 (Avoidance hitting time and avoidance fundamental matrix closed form formulation). *In an unweighted network with $\mathcal{F}$ as the avoiding set, if t is reachable from s, the avoidance hitting time and the avoidance fundamental matrix are calculated from the following closed form formulation:*

$$H_s^{\{t,\overline{\mathcal{F}}\}} = \frac{\sum_m F_{sm}^{\{t,\mathcal{F}\}} Q_m^{\{t,\overline{\mathcal{F}}\}}}{Q_s^{\{t,\overline{\mathcal{F}}\}}}, \tag{3.20}$$

$$F_{sm}^{\{t,\overline{\mathcal{F}}\}} = \frac{F_{sm}^{\{t,\mathcal{F}\}} Q_m^{\{t,\overline{\mathcal{F}}\}}}{Q_s^{\{t,\overline{\mathcal{F}}\}}}, \tag{3.21}$$

otherwise $Q_s^{\{t,\overline{\mathcal{F}}\}}$ is zero and t is not reachable from node s.

Proof.

$$\begin{aligned}
& \sum_k k[P_{\mathcal{TT}}^{k-1} P_{\mathcal{TA}}]_{st} \\
= \; & u_s'(I + 2P_{\mathcal{TT}} + 3P_{\mathcal{TT}}^2 + ...)P_{\mathcal{TA}} u_t \\
= \; & u_s'(I - P_{\mathcal{TT}})^{-2} P_{\mathcal{TA}} u_t \\
= \; & u_s' F^2 P_{\mathcal{TA}} u_t \\
= \; & u_s' F Q^{\{t,\overline{\mathcal{F}}\}} \\
= \; & \sum_m F_{sm}^{\{t,\mathcal{F}\}} Q_m^{\{t,\overline{\mathcal{F}}\}},
\end{aligned}$$

where u_i is a column vector of all 0's but its i-th entry which is equal to 1, u' is its

transpose, and $\mathcal{T} = V \setminus (\mathcal{F} \cup \{t\})$.

$$\begin{aligned}
& \sum_k [P_{\mathcal{T}\mathcal{T}}^{k-1} P_{\mathcal{T}\mathcal{A}}]_{st} \\
= \; & u_s'(I + P_{\mathcal{T}\mathcal{T}} + P_{\mathcal{T}\mathcal{T}}^2 + ...) P_{\mathcal{T}\mathcal{A}} u_t \\
= \; & u_s'(I - P_{\mathcal{T}\mathcal{T}})^{-1} P_{\mathcal{T}\mathcal{A}} u_t \\
= \; & u_s' F P_{\mathcal{T}\mathcal{A}} u_t \\
= \; & u_s' Q^{\{t,\overline{\mathcal{F}}\}} \\
= \; & Q_s^{\{t,\overline{\mathcal{F}}\}}
\end{aligned}$$

$$\Longrightarrow H_s^{\{t,\overline{\mathcal{F}}\}} = \frac{\sum_k k[P_{\mathcal{T}\mathcal{T}}^{k-1} P_{\mathcal{T}\mathcal{A}}]_{st}}{\sum_k [P_{\mathcal{T}\mathcal{T}}^{k-1} P_{\mathcal{T}\mathcal{A}}]_{st}} = \frac{\sum_m F_{sm}^{\{t,\mathcal{F}\}} Q_m^{\{t,\overline{\mathcal{F}}\}}}{Q_s^{\{t,\overline{\mathcal{F}}\}}} \tag{3.22}$$

$$\Longrightarrow F_{sm}^{\{t,\overline{\mathcal{F}}\}} = \frac{F_{sm}^{\{t,\mathcal{F}\}} \sum_k [P_{\mathcal{T}\mathcal{T}}^{k-1} P_{\mathcal{T}\mathcal{A}}]_{mt}}{\sum_k [P_{\mathcal{T}\mathcal{T}}^{k-1} P_{\mathcal{T}\mathcal{A}}]_{st}} = \frac{F_{sm}^{\{t,\mathcal{F}\}} Q_m^{\{t,\overline{\mathcal{F}}\}}}{Q_s^{\{t,\overline{\mathcal{F}}\}}} \tag{3.23}$$

□

In the theorem above, the fundamental matrix $F^{\{t,\mathcal{F}\}}$ and the absorption probability $Q^{\{t,\overline{\mathcal{F}}\}}$ are simply derived from (2.1) and (2.16) respectively: $F^{\{t,\mathcal{F}\}} = (I - P_{\mathcal{T}\mathcal{T}})^{-1}$ and $Q^{\{t,\overline{\mathcal{F}}\}} = F^{\{t,\mathcal{F}\}} P_{\mathcal{T}\mathcal{A}}$, where $\mathcal{A} = \{t\} \cup \mathcal{F}$, and $\mathcal{T}$ encompasses the rest of the nodes. Note that if set $\mathcal{F}$ is empty, Q is a column vector of all 1's and (3.20) reduces to (2.7), i.e. calculates the classical hitting time distance from s to t. The same happens for (3.21) by being reduced to classical fundamental matrix. It can be seen from (3.20) and (3.21) that the same relation of classical hitting time and classical fundamental matrix (2.7) holds for the avoidance hitting time and the avoidance fundamental matrix too:

Corollary 7.

$$H_s^{\{t,\overline{\mathcal{F}}\}} = \sum_m F_{sm}^{\{t,\overline{\mathcal{F}}\}} = F_{s:}^{\{t,\overline{\mathcal{F}}\}} \mathbf{1}, \tag{3.24}$$

where $F_{s:}^{\{t,\overline{\mathcal{F}}\}}$ is the s-th row of avoidance fundamental matrix.

Theorem 4 (Avoidance hitting cost closed form formulation)**.** *In a weighted network with transition probability P, weight (cost) matrix W, and $\mathcal{F}$ as the avoiding set, if t is*

reachable from s, the avoidance hitting cost is calculated from the following closed form formulation:

$$U_s^{\{t,\overline{\mathcal{F}}\}} = \frac{\sum_m F_{sm}^{\{t,\mathcal{F}\}} Q_m^{\{t,\overline{\mathcal{F}}\}} r_m^{\{t,\overline{\mathcal{F}}\}}}{Q_s^{\{t,\overline{\mathcal{F}}\}}}, \tag{3.25}$$

where $r_m^{\{t,\overline{\mathcal{F}}\}} = \sum_j \frac{Q_j^{\{t,\overline{\mathcal{F}}\}}}{Q_m^{\{t,\overline{\mathcal{F}}\}}} P_{mj} w_{mj}$. *If t is not reachable from s, $Q_s^{\{t,\overline{\mathcal{F}}\}}$ is zero.*

Proof. For transient set equal to $\mathcal{T} = V \setminus \mathcal{F} \cup \{t\}$ we have:

$$\begin{aligned}
U_s^{\{t,\overline{\mathcal{F}}\}} &= \frac{\sum_{\zeta_j \in Z_{st}^{\overline{\mathcal{F}}}} l_{\zeta_j} \Pr_{\zeta_j}}{\sum_{\zeta_j \in Z_{st}^{\overline{\mathcal{F}}}} \Pr_{\zeta_j}} \\
&= \frac{\sum_{\zeta_j \in Z_{st}^{\overline{\mathcal{F}}}} \Pr_{\zeta_j} \sum_{k=1}^{k_{\zeta_j}} w_{v_{k-1}v_k}}{Q_s^{t,\overline{\mathcal{F}}}} \\
&= \frac{\sum_{\zeta_j \in Z_{st}^{\overline{\mathcal{F}}}} \sum_{k=1}^{k_{\zeta_j}} [\prod_{i=1}^{k} P_{v_{i-1}v_i} . (P_{v_k v_{k+1}} w_{v_k v_{k+1}}) . \prod_{i=k+2}^{k_{\zeta_j}} P_{v_{i-1}v_i}]}{Q_s^{t,\overline{\mathcal{F}}}} \\
&= \frac{\sum_{e_{xy} \in E, x \in \mathcal{T}, y \in \mathcal{T} \cup \{t\}} P_{xy} w_{xy} (\sum_{\zeta_j \in Z_{sx}^{\overline{\mathcal{F}}}} \Pr_{\zeta_j}) . (\sum_{\zeta_i \in Z_{yt}^{\overline{\mathcal{F}}}} \Pr_{\zeta_i})}{Q_s^{t,\overline{\mathcal{F}}}} \\
&= \frac{\sum_{e_{xy} \in E, x \in \mathcal{T}, y \in \mathcal{T} \cup \{t\}} P_{xy} w_{xy} (\sum_k \sum_{\zeta_j \in Z_{sx}^{\overline{\mathcal{F}}}(k)} \Pr_{\zeta_j}) . (\sum_k \sum_{\zeta_i \in Z_{yt}^{\overline{\mathcal{F}}}(k)} \Pr_{\zeta_i})}{Q_s^{t,\overline{\mathcal{F}}}} \\
&= \frac{\sum_{e_{xy} \in E, x \in \mathcal{T}, y \in \mathcal{T} \cup \{t\}} P_{xy} w_{xy} (\sum_k [P_{\mathcal{T}\mathcal{T}}^k]_{sx}) . (\sum_k [P_{\mathcal{T}\mathcal{T}}^{k-1} P_{\mathcal{T}\mathcal{A}}]_{yt})}{Q_s^{t,\overline{\mathcal{F}}}} \\
&= \frac{\sum_{e_{xy} \in E, x \in \mathcal{T}, y \in \mathcal{T} \cup \{t\}} P_{xy} w_{xy} F_{sx}^{\{t,\mathcal{F}\}} Q_y^{\{t,\overline{\mathcal{F}}\}}}{Q_s^{t,\overline{\mathcal{F}}}} \\
&= \frac{\sum_{x \in \mathcal{T}} F_{sx}^{\{t,\mathcal{F}\}} Q_x^{\{t,\overline{\mathcal{F}}\}} \sum_{y \in \mathcal{N}_{out}(x) \setminus \mathcal{F}} p_{xy} \frac{Q_y^{\{t,\overline{\mathcal{F}}\}}}{Q_x^{\{t,\overline{\mathcal{F}}\}}} w_{xy}}{Q_s^{\{t,\overline{\mathcal{F}}\}}} \\
&= \frac{\sum_{x \in \mathcal{T}} F_{sx}^{\{t,\mathcal{F}\}} Q_x^{\{t,\overline{\mathcal{F}}\}} r_x^{\{t,\overline{\mathcal{F}}\}}}{Q_s^{\{t,\overline{\mathcal{F}}\}}}
\end{aligned} \tag{3.26}$$

□

Note that the avoidance fundamental matrix is the same as the unweighted case. For understanding the notations please refer to Table (1.1).

Corollary 8.

$$U_s^{\{t,\overline{\mathcal{F}}\}} = \sum_m F_{sm}^{\{t,\overline{\mathcal{F}}\}} r_m^{\{t,\overline{\mathcal{F}}\}} = F_{s:}^{\{t,\overline{\mathcal{F}}\}} \boldsymbol{r}^{\{t,\overline{\mathcal{F}}\}}, \tag{3.27}$$

Theorem 5 (Generalization of avoidance fundamental matrix and avoidance hitting time for a set of targets)**.** *Let $T = \{t_1, ..., t_c\}$ be a set of targets. The expected number of steps to hit and the expected number of visits to m before hitting any state in T avoiding set $\mathcal{F}$ are computed from the following formulation respectively:*

$$H_s^{\{T,\overline{\mathcal{F}}\}} = \frac{\sum_m F_{sm}^{\{T,\mathcal{F}\}} Q_m^{\{T,\overline{\mathcal{F}}\}}}{Q_s^{\{T,\overline{\mathcal{F}}\}}}, \tag{3.28}$$

$$F_{sm}^{\{T,\overline{\mathcal{F}}\}} = \frac{F_{sm}^{\{T,\mathcal{F}\}} Q_m^{\{T,\overline{\mathcal{F}}\}}}{Q_s^{\{T,\overline{\mathcal{F}}\}}}, \tag{3.29}$$

where $Q_m^{\{T,\overline{\mathcal{F}}\}}$ is the summation of absorption probabilities over the columns corresponding to set T.

The proof is very similar to proof of Theorem (3) and we do not repeat it here.

Lemma 4. *Decomposing the classical hitting time for two target nodes in terms of avoidance hitting time yields:*

$$H_s^{\{t,k\}} = Q_s^{\{t,\bar{k}\}} H_s^{\{t,\bar{k}\}} + Q_s^{\{k,\bar{t}\}} H_s^{\{k,\bar{t}\}} \tag{3.30}$$

Proof. This is proved according to (2.14) for $T = \{t, k\}$, (3), and (2.54). □

Theorem 6 (Hitting Time Decomposition)**.** *The hitting time from node s to node t can be decomposed into an "avoidance" hitting time component and a "transit" hitting time component with respect to any node k as follows:*

$$H_s^{\{t\}} = Q_s^{\{t,\bar{k}\}} H_s^{\{t,\bar{k}\}} + Q_s^{\{k,\bar{t}\}} H_s^{\{t,\check{k}\}}. \tag{3.31}$$

Proof. Taking sum over m for both sides of Eq. (2.50) and substituting $\frac{F_{ik}^{\{j\}}}{F_{kk}^{\{j\}}}$ by $Q_i^{\{k,\bar{j}\}}$ (Theorem (2)), the following equation is obtained:

$$H_i^{\{j,k\}} = H_i^{\{j\}} - Q_i^{\{k,\bar{j}\}} H_k^{\{j\}} \tag{3.32}$$

Substituting Lemma (4), which is $H_i^{\{j,k\}} = Q_i^{\{k,\bar{j}\}} H_i^{\{k,\bar{j}\}} + Q_i^{\{j,\bar{k}\}} H_i^{\{j,\bar{k}\}}$, in Eq. (3.32) yields the following relation:

$$H_i^{\{j\}} = Q_i^{\{j,\bar{k}\}} H_i^{\{j,\bar{k}\}} + Q_i^{\{k,\bar{j}\}} (H_i^{\{k,\bar{j}\}} + H_k^{\{j\}}) = Q_i^{\{j,\bar{k}\}} H_i^{\{j,\bar{k}\}} + Q_i^{\{k,\bar{j}\}} H_i^{\{j,\check{k}\}},$$

where $H_i^{\{j,\check{k}\}} = H_i^{\{k,\bar{j}\}} + H_k^{\{j\}}$.

□

The transit and avoidance hitting times can be generalized to an arbitrary (sub)set of nodes, $H_s^{\{t,\check{S}_1\}}$, $H_s^{\{t,\bar{S}_2\}}$, and combined, $H_s^{\{t,\check{S}_1,\bar{S}_2\}}$, where the last term represents the hitting time from node s to node t conditioned on traversing any node in S_1 while avoiding all nodes in S_2.

Theorem 7. *(Avoidance Paradigm to Classical Paradigm Transformation)*
Network G with avoiding node o and target set T can be transformed to network $\acute{G}$ without node o and target set T such that the avoidance metrics in the former network turn into the classical metrics in the latter network, i.e. $F_{sm}^{\{T,\bar{o}\}} = \acute{F}_{sm}^{\{T\}}$, $H_s^{\{T,\bar{o}\}} = \acute{H}_s^{\{T\}}$, and $U_s^{\{T,\bar{o}\}} = \acute{U}_s^{\{T\}}$. The transformation function between transition matrix $\acute{P}$ belonging to $\acute{G}$ and P belonging to G is as follows:

$$\acute{P}_{ij} = P_{ij} \frac{Q_j^{\{T,\bar{o}\}}}{Q_i^{\{T,\bar{o}\}}} \tag{3.33}$$

Proof. We first prove that $\acute{P}$ is a transition probability matrix, namely is row stochastic:

$$\sum_{j \in \mathcal{N}(i)} \acute{P}_{ij} = \sum_{j \in \mathcal{N}(i)} P_{ij} \frac{Q_j^{\{T,\bar{o}\}}}{Q_i^{\{T,\bar{o}\}}} = \frac{1}{Q_i^{\{T,\bar{o}\}}} \sum_{j \in \mathcal{N}(i)} P_{ij} Q_j^{\{T,\bar{o}\}} = \frac{Q_i^{\{T,\bar{o}\}}}{Q_i^{\{T,\bar{o}\}}} = 1, \tag{3.34}$$

where the third equality is resulted because of Q is a harmonic function. Now we show that with the transformation in eq. (3.33) these equalities hold: $F_{sm}^{\{T,\bar{o}\}} = \acute{F}_{sm}^{\{T\}}$,

$H_s^{\{T,\bar{o}\}} = \mathrm{H}_s^{\{T\}}$, and $U_s^{\{T,\bar{o}\}} = \mathrm{U}_s^{\{T\}}$.

$$\begin{aligned} F^{\{T\}} &= \sum_{k=0} \mathrm{P}_{\mathcal{TT}}^k = \sum_{k=0} (Diag(Q^{T,\bar{o}})^{-1} P_{\mathcal{TT}} Diag(Q^{T,\bar{o}}))^k \\ &= \sum_{k=0} Diag(Q^{T,\bar{o}})^{-1} P_{\mathcal{TT}}^k Diag(Q^{T,\bar{o}}) \\ &= Diag(Q^{T,\bar{o}})^{-1} (\sum_{k=0} P_{\mathcal{TT}}^k) Diag(Q^{T,\bar{o}}) \\ &= Diag(Q^{T,\bar{o}})^{-1} F^{\{T,o\}} Diag(Q^{T,\bar{o}}) \\ &= F^{\{T,\bar{o}\}} \end{aligned}$$

For the hitting times we have $\mathrm{H}_s^{\{T\}} = \mathrm{F}^{\{T\}}\mathbf{1}$ and $H_s^{\{T,\bar{o}\}} = F^{\{T,\bar{o}\}}\mathbf{1}$, so $H_s^{\{T,\bar{o}\}} = \mathrm{H}_s^{\{T\}}$. The following relations also hold for hitting costs:

$$\begin{aligned} U_s^{\{T,\bar{o}\}} &= \sum_m F_{sm}^{\{T,\bar{o}\}} r_m^{\{T,\bar{o}\}} \\ &= \sum_m F_{sm}^{\{T,\bar{o}\}} \sum_j \frac{Q_j^{\{T,\bar{o}\}}}{Q_m^{\{T,\bar{o}\}}} P_{mj} w_{mj} \\ &= \sum_m F_{sm}^{\{T,\bar{o}\}} \sum_j P_{mj} w_{mj} \\ &= \sum_m F_{sm}^{\{T,\bar{o}\}} \mathrm{r}_m \\ &= \sum_m \mathrm{F}_{sm}^{\{T\}} \mathrm{r}_m \\ &= \mathrm{U}_s^{\{T\}}, \end{aligned}$$

where the first and third equalities are based on (3.27) and (3.33) respectively. □

Relation 10. *For a fixed avoiding node k, the following formulation is useful to find the avoidance hitting time for any pairs of source s and target t from matrix $F^{\{k\}}$:*

$$H_s^{\{t,\bar{k}\}} = \sum_m F_{mt}^{\{k\}} \left(\frac{F_{sm}^{\{k\}}}{F_{st}^{\{k\}}} - \frac{F_{tm}^{\{k\}}}{F_{tt}^{\{k\}}}\right) \tag{3.35}$$

Proof. Use (3) and (1). □

Relation 11. *For a fixed target node t, the following formulation is useful to find the*

avoidance hitting time for any pairs of source s and avoiding node k from matrix $F^{\{t\}}$:

$$\begin{aligned} H_s^{\{t,\bar{k}\}} &= \frac{1}{F_{kk}^{\{t\}}} \frac{\sum_m ((F_{kk}^{\{t\}} F_{sm}^{\{t\}} - F_{sk}^{\{t\}} F_{km}^{\{t\}})(F_{kk}^{\{t\}} - F_{mk}^{\{t\}}))}{F_{kk}^{\{t\}} - F_{sk}^{\{t\}}} \\ &= \frac{1}{F_{kk}^{\{t\}} - F_{sk}^{\{t\}}} (F_{kk}^{\{t\}} H_s^{\{t\}} - F_{sk}^{\{t\}} H_k^{\{t\}} - \sum_m F_{sm}^{\{t\}} F_{mk}^{\{t\}} + Q_s^{\{k,\bar{t}\}} \sum_m F_{km}^{\{t\}} F_{mk}^{\{t\}}) \end{aligned}$$

Proof. Use (3) and (1). □

Relation 12. *Avoidance fundamental matrix in terms of $F^{\{k\}}$:*

$$F_{sm}^{\{t,\bar{k}\}} = F_{mt}^{\{k\}} (\frac{F_{sm}^{\{k\}}}{F_{st}^{\{k\}}} - \frac{F_{tm}^{\{k\}}}{F_{tt}^{\{k\}}}) \tag{3.36}$$

Proof. Use (3) and (1) □

Relation 13. *Avoidance fundamental matrix in terms of $F^{\{t\}}$:*

$$\begin{aligned} F_{sm}^{\{t,\bar{k}\}} &= \frac{1}{F_{kk}^{\{t\}}} \frac{(F_{kk}^{\{t\}} F_{sm}^{\{t\}} - F_{sk}^{\{t\}} F_{km}^{\{t\}})(F_{kk}^{\{t\}} - F_{mk}^{\{t\}})}{F_{kk}^{\{t\}} - F_{sk}^{\{t\}}} \\ &= \frac{F_{kk}^{\{t\}} F_{sm}^{\{t\}} - F_{sk}^{\{t\}} F_{km}^{\{t\}} - F_{sm}^{\{t\}} F_{mk}^{\{t\}} + Q_s^{\{k,\bar{t}\}} F_{km}^{\{t\}} F_{mk}^{\{t\}}}{F_{kk}^{\{t\}} - F_{sk}^{\{t\}}} \end{aligned} \tag{3.37}$$

Proof. Use (3) and (1). □

Part III

Cascade

Chapter 4

Influence Maximization in Social Networks

4.1 Introduction

Motivated by viral marketing and other applications, the problem of influence maximization in a social network has attracted much attention in recent years. Given a social network where nodes represent users in a social group, and edges represent relationships and interactions between the users (and through which they influence each other), the basic idea of influence maximization is to select an initial set of "most influential" users (often referred to as the *seeds*) among all users so as to maximize the total influence under a given diffusion process (often referred to as the *influence model*) on the social network. In the context of viral marketing, this amounts to by initially targeting a set of influential customers, e.g., by providing them with free product samples, with the goal to trigger a cascade of influence through "word-of-mouth" or recommendations to friends to maximize the total number of customers adopting the said product. Domingos and Richardson [40] introduced this algorithmic problem to the Computer Science community and Kempe et al. [74] made the topic vastly popular under the name of *influence maximization*. They studied two influence models, the independent cascade (IC) model and the linear threshold (LT) model, and applied a greedy method to tackle the influence maximization problem [74]. Unfortunately Kempe et al.'s approach [74] for calculating the influence spread is based on Monte Carlo simulations which does not

scale to large networks [31, 30]. As the result, it motivated researchers to either improve the scalability [31, 30] or study more tractable influence models [63, 43].

The focus of almost all of these earlier studies are, however, *progressive* influence models, including LT and IC models, in which once a costumer adopts a product or a user performs an action she cannot revert it. Retweeting news and sharing videos in online social network websites, are examples of progressive, i.e. irreversible actions. Nevertheless, there are numerous real world instances where the actions are *non-progressive* especially in technology adoption domain. For example, adopting a cell phone service provider, such as AT&T and T-mobile, is a non-progressive action where a user can switch between providers. The objective of influence maximization in this example is to persuade more users to adopt the intended provider for a longer period of time. To capture the reversibility of choices in real scenarios, we present Heat Conduction (HC) model which has favorable real-world interpretation. We also show that HC unifies, generalizes, and extends the existing non-progressive models, including non-progressive LT (NLT) [74] and Voter model [48] (see Section 6.5). In contrast to the Voter model, HC does not *necessarily* reach consensus, where one product dominates and extinguishes the others after finite time, so the proposed HC model can explain the *coexistence of multiple product adoptions*, which is a typical phenomena in real world. In addition, HC model incorporates both "social" and "non-social" factors, e.g., intrinsic inertia or reluctance of some users in adopting a new idea or trying out a new product, external "media effect" which exerts a "non-social" influence in promoting certain ideas or products.

We tackle the influence maximization problem under HC influence model with a *scalable* and provably *near-optimal* solution. Kempe et al.'s approach [74] for influence maximization under NLT model, is to reduce the model to (progressive) LT by replicating the network as many as time progresses and compute the influence spread by the same slow Monte Carlo method for the resulted huge network. This approach is practically impossible for large networks, specially for the *infinite time horizon*. We also prove that contrary to the Voter, for which the influence maximization can be solved *exactly* in polynomial time [48], the influence maximization for HC is NP-hard. We develop an approximation (greedy) algorithm for influence maximization under HC for infinite time horizon with guaranteed *near-optimal* performance. Exploiting probability theory and

novel Markov chain metrics, we are able to provide *closed form* solution for both computing the influence spread and greedy selection step which entirely removes the need to explicitly evaluate each node as the best seed candidate; our fast and scalable algorithm, C2Greedy, for influence maximization under HC removes the computational barrier that prevented the literature from considering the non-progressive influence models.

Our extensive experiments on several and large real and synthetic networks validate the efficiency and effectiveness of our method which outperforms the state-of-the-art in terms of both influence spread and scalability; we show that the most influential nodes under progressive models not necessarily act as the most influentials under non-progressive models and a *designated* non-progressive algorithm is necessary. Moreover, we present the first real non-progressive cascade dataset which models the non-progressive propagation of research topics among network of researchers. We are planning to make this data publicly available. Our contribution in this work is summarized as follows:

• We propose HC influence model that has favorable real world interpretations, and unifies, generalizes, and extends the existing non-progressive models and .

• We show HC has three noble key properties which enables us solving influence maximization efficiently.

• To the best of our knowledge, we are the first to present a scalable solution for influence maximization under non-progressive LT model.

• We demonstrate high performance and scalability of our algorithm via extensive experiments and present the first ever real non-progressive cascade dataset.

The rest of this chapter is organized as follows. After a brief review on the related work, we introduce our HC model in Section 4.3. Next, we show how to compute the influence spread for HC in closed form in Section 4.4. In Section 4.5, we present our efficient algorithm C2Greedy for influence maximization under the HC model. Section 6.5 explains how HC unifies other non-progressive models and provides a more complete view of the HC model. Finally we conduct comprehensive experiments in Section 4.7 to illustrate performance of our algorithm.

4.2 Related work

After the debut of influence maximization as a data mining problem [40], it is formulated as a discrete optimization problem based on progressive influence models (LT and IC) from social and physical sciences [74]. Kempe et al. [74] show that influence maximization is NP-hard under LT and IC models but the influence spread is submodular for the models which enables them to use the greedy method. Although the algorithm is greedy it usually does not scale, because it needs to compute influence spread many times in each iteration while influence spread has no known closed form and is estimated by Monte Carlo simulation. The follow-up studies [89, 31, 30, 65, 63, 43] attempt to speed up this process by avoiding or decreasing the need for the MC simulation (for further details of the studies on progressive influence model please refer to Supplementary). Kempe et al. [74] also introduce a non-progressive version of the LT influence model (NLT) and try to tackle the influence maximization problem under NLT by reducing the model to (progressive) LT, discussed in Section 4.1.
Voter model, as the most well-known non-progressive model, is originally introduced in [34, 68] and adopted for viral marketing in [48]. Even-Dar and Shapira show that under Voter model, highest degree nodes are the solution of influence maximization [48]. Unfortunately since the Voter model reaches consensus, i.e. one product remains in long term, it can not explain the coexistence of multiple product adoptions, which is a typical case in many real product adoptions.

• **Influence maximization under progressive model: A brief review**
CELF method of Leskovec et al. [89] attempts to speed up the original greedy method, proposed by Kempe et al. [74], by reducing the number of calls to Monte Carlo routine for spread computation. CELF lazy method is based on the submodularity of the influence spread and can be applied to any submodular maximization problem. Although lazy evaluation improves the running time of the original greedy method by up to 700 times [89], it still does not scale to large graphs [31].

Recently heuristics have been proposed to approximate influence spread for LT [31] and IC [30] which enables the greedy method to scale for large networks. Chen et al. [31] suggest to use a local directed acyclic graph (LDAG) per node, instead of considering the whole graph, to approximate the influence flowing to the node. Goyal et

al. propose SIMPATH method [65] under the LT model which is built on CELF method [89]. They approximate the influence spread by enumerating the simple paths starting from the seeds within a small neighborhood. Both of these methods have parameters to be tuned which control the trade-off between running time and accuracy of influence spread estimation. Methods presented in [31, 65] accelerate the greedy method [74] substantially and achieve high performance in influence maximization.

Gomez-Rodriguez et al. [63] propose a progressive continuous time influence model with dynamics similar to IC and show that influence maximization is NP-hard for this model as well. They show submodularity of influence spread and exploit the same greedy algorithm. In contrast to all other progressive models, influence spread has a closed form for this model but the computation is not scalable for large scale networks. A recent work [43] has scaled influence computation by developing a randomized algorithm for approximating it.

4.3 Heat Conduction Influence Model

The heat conduction (HC) influence model is inspired by the resemblance of influence diffusion through a social network to heat conduction through an object, where heat is transferred from the part with higher temperature to the part with lower temperature. We provide a simple description of HC in this section and defer the complete view of it as well as its unification property to Section 6.5.

Considering directed graph $G = (\mathcal{V}, \mathcal{E})$ which represents the social (influence) network, the directed edge from node i to node j declares that i follows j (or equivalently j influences i). Edge weight ω_{ij} indicates the amount that i trusts j and unless specified $0 \leq \omega_{ij} \leq 1$. The set of i's neighbors, representing the nodes that influence i, is denoted by $\mathcal{N}(i)$. The influence cascade can be assumed as a *binary* process in which a node who adopts the "desired" product is called *active*, and *inactive* otherwise. Note that this assumption holds for the cases with multiple products as well, where the objective is to maximize the influence (publicity) of the "desired" product, and the rest are all considered "undesired". *Seed* is a node that has been selected for the direct marketing and remains active during the entire process. In HC model, the influence cascade is initiated from a set of seeds S and arbitrary values for other nodes. The *choice* of node

i to become active or inactive at time $t+1$ is a linear function of the choices of its neighbors at time t as well as its intrinsic (or non-social) bias toward activeness:

$$Pr\big(\delta_i(t+1)=1|\mathcal{N}(i)\big)=\beta_i b+(1-\beta_i)\sum_{j\in\mathcal{N}(i)}\omega_{ij}\delta_j(t), \tag{4.1}$$

where $\beta_i \in (0,1)$, $b\in[0,1]$, and $\sum_{j\in\mathcal{N}(i)}\omega_{ij}=1$. Indicator function $\delta_i(t)$ is 1 when node i adopts the desired product at time t and 0 otherwise. We refer to (4.1) as the *choice rule*. The dependence on neighbors in (4.1) represents the "social" influence and the bias value b accounts for "non-social" influence which comes from any source out of the neighbors, e.g. media. The "non-social" influence can explain the cases where the "social" influence alone fails to model the cascades [25]. We discuss further interpretation and extensions of HC in Section 6.5.

Replacing the choice rule (4.1) in $Pr\big(\delta_i(t+1)\big)=\sum Pr(\delta_i(t+1)|\mathcal{N}(i))Pr(\mathcal{N}(i))$ results in the following *probabilistic* interpretation of the original binary HC model. Each node i has a value at time t denoted by $u(i,t)$ which represents the *probability* that she adopts the desired product at time t:

$$u(i,t+1)=\beta_i b+(1-\beta_i)\sum_{j\in\mathcal{N}(i)}\omega_{ij}u(j,t), \tag{4.2}$$

Simple calculation shows that the bias value b can be integrated into the network by adding a bias node n (assuming that the network has $n-1$ nodes) with adoption probability b. Therefore, HC dynamics converts to the following:

$$u(i,t+1)=\sum_{j\in\mathcal{EN}(i)}P_{ij}u(j,t), \tag{4.3}$$

where $\mathcal{EN}(i)=\mathcal{N}(i)\cup\{n\}$ is the extended neighborhood, $P_{in}=\beta_i$, $u(n,t)=b$, and $\forall j\neq n: P_{ij}=(1-\beta_i)\omega_{ij}$. Rewriting (4.3) in the following form shows that HC follows the discrete form of **Heat Equation** [82], which reveals the naming reason of HC influence model: $u(:,t+1)-u(:,t)=(P-I)u(:,t)$, where $\mathcal{L}=I-P$ is the Laplacian matrix, $u(i,t)$ is the temperature of particle i at time t, and ":" denotes the vector of all entries.

4.4 HC Influence Spread

Influence spread of set $\mathcal{S}$ for time t is defined as the expected number of active nodes at time t of a cascade started with $\mathcal{S}$. Knowing that $u(i,t)$ is the probability of node i being active at time t, *influence spread* (or function) $\sigma(\mathcal{S},t)$ is computed from:

$$\sigma(\mathcal{S},t) = \sum_{i\in\mathcal{V}} u(i,t). \tag{4.4}$$

Motivated by the classical heat transfer methods, the initial and the boundary conditions should be specified to solve the heat equation and find $u(i,t)$ uniquely. In HC, the seeds $\mathcal{S}$ and the bias node are the boundary nodes and the rest are interiors. Assuming $\mathcal{S} = \{n-1, n-2, ..., n-|\mathcal{S}|\}$ and n as the bias node, HC is defined by the following heat equation system:

$$\begin{aligned}
\text{Main equation} \quad &: \quad u(:,t+1) - u(:,t) = -\mathcal{L}u(:,t) \\
\text{Boundary conditions} \quad &: \quad u(n,t) = b, \\
&\quad\quad u(s,t) = 1 \quad \forall s \in \mathcal{S} \\
\text{Initial condition} \quad &: \quad u(:,0) = z + [0, ..., 0, \underbrace{1, ..., 1}_{|\mathcal{S}|}, b]',
\end{aligned} \tag{4.5}$$

where, as indicated in this formula, initial value $u(:,0)$ is the sum of two vectors: the initial values of the interior nodes (z) and the initial values of boundaries (the second vector). The corresponding entries of boundaries in z are zero. In the continue, exploiting probability theory and novel Markov chain metrics, we provide a closed form solution to this heat equation system.

Social network G can be interpreted as an absorbing Markov chain where the absorbing states (boundary set $\mathcal{B}$) are the seeds and bias node, $\mathcal{B} = \mathcal{S} \cup \{n\}$, and P_{ij} is the probability of transition from i to j. The adoption probability of the nodes at time t, i.e. $u(:,t)$, can be written as a linear function of initial condition (4.3):

$$u(:,t) = P^t u(:,0), \tag{4.6}$$

where P is row-stochastic and has the following block form: $P = \begin{bmatrix} R & B \\ \mathbf{0} & I \end{bmatrix}$. The superscript indicates the time here. The boundary set by definition have fixed values

over time and do not follow any other nodes which leads to the zero and identity blocks $I_{(|\mathcal{S}|+1)\times(|\mathcal{S}|+1)}$. Blocks R and B represent transition probabilities of interior-to-interior and interior-to-boundary respectively. Note that different boundary conditions in (4.5), like different seed set, result in a different P. Therefore both P and $u(:,t)$ implicitly depend on $\mathcal{S}$.

When t goes to infinity, transient part of u vanishes and it converges to the steady-state solution $v = u(:,\infty)$, which is independent of time and is Harmonic, meaning that it satisfies $Pv = v$ [41]. Assume $v = \big(v_{\mathcal{I}}, v_{\mathcal{B}}\big)^T$ where $\mathcal{I} = \mathcal{V} \setminus \mathcal{B}$ is the set of interior nodes, then the value of interior nodes is computed from boundary nodes [41]:

$$v_{\mathcal{I}} = (I - R)^{-1} B v_{\mathcal{B}} = F B v_{\mathcal{B}} = Q v_{\mathcal{B}}. \tag{4.7}$$

where $F = (I - R)^{-1}$ is the *fundamental matrix* and F_{ij} indicates the average number of times that a random walk started from i passes j before absorption by any absorbing (boundary) nodes [41]. Also, the *absorption probability* matrix $Q = FB$ is a $(n - |\mathcal{S}| - 1) \times (|\mathcal{S}| + 1)$ row-stochastic matrix, where Q_{ij} denotes the probability of absorption of a random walk started from i by the absorbing node j [41].

From here on, without loss of generality, we assume b to be zero in equation (4.5). Using (4.6) and (4.7), the influence spreads for infinite time can be computed in closed form:

$$\sigma(\mathcal{S}, \infty) = \sum_{i=1}^{n} v(i) = |\mathcal{S}| + \sum_{i \in \mathcal{I}} \sum_{s \in \mathcal{S}} Q^{\mathcal{S}}_{is}. \tag{4.8}$$

The superscript in $Q^{\mathcal{S}}$ and $P^{\mathcal{S}}$ explicitly indicates that they are functions of seed set $\mathcal{S}$. Note that in fact they are depending on the total boundary set, $\mathcal{B} = \mathcal{S} \cup \{n\}$, but since the bias node is always a boundary, throughout this chapter we discard it from the superscripts to avoid clutter.

4.5 Influence Maximization for HC

In this section we solve the influence maximization problem for *infinite time horizon* under HC model, formulated as follows:

$$\mathcal{S}^* = \arg\max_{\mathcal{S} \subseteq \mathcal{V}} \sigma(\mathcal{S}, \infty), \qquad s.t. \qquad |\mathcal{S}| \leq K. \tag{4.9}$$

4.5.1 Influence Maximization for $K = 1$

Based on (4.8) and (4.9), the most influential person (MIP) is the solution of the following optimization problem: $\arg\max_{\mathcal{V}\setminus\{n\}} \sum_{i\in\mathcal{V}\setminus\{s,n\}} Q_{is}^{\{s\}}$. This equation states that to find the MIP, we need to pick each candidate s and make it absorbing and compute the new P as $P^{\{s\}}$ which in turn changes Q to $Q^{\{s\}}$, and repeat this procedure $n-1$ times for all s. This procedure is problematic because for each $Q^{\{s\}}$ we require to recompute matrix $F^{\{s\}}$ which involves matrix inversion. But, in the following theorem we show that we are able to do this by only one matrix inversion instead of $n-1$ matrix inversions, and having matrix $F^{\emptyset}$ is enough to find the most influential person of the network ($\emptyset$ sign indicated no seed is selected):

Theorem 8. *MIP under HC (4.1) when $t \to \infty$ can be computed in closed form from the following formula:*

$$MIP = \arg\max_{s\in\mathcal{V}\setminus\{n\}} \sum_{i\in\mathcal{V}\setminus\{n\}} \frac{F_{is}^{\emptyset}}{F_{ss}^{\emptyset}} = \arg\max \mathbf{1}'\breve{F}^{\emptyset}, \tag{4.10}$$

where $\breve{F}^{\emptyset}$ is $F^{\emptyset}$ when each of its columns is normalized by the corresponding diagonal entry. Note that left multiplication of all ones row vector is just a column-sum operation.

4.5.2 Influence Maximization for $K > 1$

Although the influence maximization can be solved optimally for $K = 1$, the general problem (4.9) under HC for $K > 1$ is NP-hard:

Theorem 9. *Given a network $G = (\mathcal{V}, \mathcal{E})$ and a seed set $\mathcal{S} \subseteq \mathcal{V}$, influence maximization for infinite time horizon (4.9) under HC defined by (4.1) is NP-hard.*

In spite of being NP-hard, we show that the influence spread $\sigma(\mathcal{S}, \infty)$ is *submodular* in the seed set $\mathcal{S}$ which enables us to find a provable near-optimal greedy solution. A set function $f : 2^{\mathcal{V}} \to \mathbb{R}$ maps subsets of a finite set $\mathcal{V}$ to the real numbers and is submodular if for $\mathcal{T} \subseteq \mathcal{S} \subseteq \mathcal{V}$ and $s \in \mathcal{V}\setminus\mathcal{S}$, $f(\mathcal{T}\cup\{s\}) - f(\mathcal{T}) \geq f(\mathcal{S}\cup\{s\}) - f(\mathcal{S})$ holds, which is the diminishing return property. Following theorem presents our established submodularity results.

Theorem 10. *Given a network $G = (\mathcal{V}, \mathcal{E})$, influence spread $\sigma(\mathcal{S}, \infty)$ under HC model is non-negative monotone submodular function.*

The greedy solution adds nodes to the seed set $\mathcal{S}$ sequentially and maximizes a monotone submodular function with $(1-1/e)$ factor approximation guarantee [95]. More formally the $(k+1)$-th seed is the node with maximum **marginal gain**: $(k+1)$th-$MIP_t = \arg\max_{s\in\mathcal{V}\setminus\{\mathcal{S}_k\cup\{n\}\}} \sigma(\mathcal{S}_k\cup\{s\},t) - \sigma(\mathcal{S}_k,t)$, where $\mathcal{S}_k$ is the set of k seeds which have been picked already. Although we can compute the above objective function in closed form, for selecting the next seed we have to test all s to solve the problem which is the approach of all existing greedy based method in the literature. Previously a lazy greedy scheme have been introduced to reduce the number testing candidate nodes s [89]. In the next section we go one step further and show that under HC model and for *infinite time horizon* we can solve the marginal gain in *closed form*.

4.5.3 Greedy Selection

An important characteristic of the linear systems, like HC when $t\to\infty$, is the "superposition" principle. We leverage this principle to calculate the marginal gain of the nodes efficiently and pick the one with maximum gain for the greedy algorithm. Based on this principle, the value of each node in HC for infinite time, and for a given seed set $\mathcal{S}$, is equal to the algebraic sum of the values caused by each seed acting alone, while all other values of seeds have been kept zero. Therefore, when a node s is added to the seed set $\mathcal{S}_k$, its marginal gain can be calculated as the summation of values of the nodes when all of the values of $\mathcal{S}_k$ have been turned to zero and node s is the only seed in the network, whose value is $1-v^{\mathcal{S}_k}(s)$. In this new problem, the vector of boundary values $v_{\mathcal{B}}^{\mathcal{S}_k\cup\{s\}}$ is a vector of all 0's except the entry corresponding to the node s with value $1-v^{\mathcal{S}_k}(s)$, and the value of interior node i is obtained from (4.7):

$$v_{\mathcal{I}}^{\mathcal{S}_k\cup\{s\}}(i) = Q_{is}^{\mathcal{S}_k\cup\{s\}}(1-v^{\mathcal{S}_k}(s))$$

Substituting Q from lemma 3 result (see Supplementary), the $k+1$-th seed is determined from the following closed form equation:

$$\begin{aligned} & (k+1)\text{th-}MIP \\ = & \arg\max_{s\in\mathcal{V}\setminus\{\mathcal{S}_k\cup\{n\}\}} \sum_{i\in\mathcal{V}\setminus\{\mathcal{S}_k\cup\{n\}\}} \frac{F_{is}^{\mathcal{S}_k}}{F_{ss}^{\mathcal{S}_k}}\big(1-v^{\mathcal{S}_k}(s)\big), \\ = & \arg\max(\mathbf{1}-v^{\mathcal{S}_k})'\breve{F}^{\mathcal{S}_k} \end{aligned} \tag{4.11}$$

Note that vector $v^{\mathcal{S}_k}$ is obtained in step k and is known, and matrix $F^{\mathcal{S}_k}$ can be calculated from $F^{\mathcal{S}_{k-1}}$ without any need for matrix inversion (see Supplementary, lemma 1). One may observe that equation (4.11) is the general form of Theorem 8, since $v^{\mathcal{S}_0} = v^{\emptyset} = 0$. Notice that equation (4.11) intuitively uses two criteria for selecting the new seed: its current value should be far from 1 (higher value for $(1-v^{\mathcal{S}_k}(s))$ term) which suggests that it is far from the previously selected seeds, and at the same time it should have a high network centrality (corresponding to the $F_{is}^{\mathcal{S}_k}/F_{ss}^{\mathcal{S}_k}$ term). Algorithm 1 summarizes our C2Greedy method for $t \to \infty$: a greedy algorithm with 2 closed form steps. Operator $\otimes$ in step 10 denotes the Hadamard product.

Algorithm 1 C2Greedy

input: extended directed network $G = (\mathcal{V}, \mathcal{E})$ with bias node n, maximum budget K.
output: seed set $\mathcal{S}_K \subseteq \mathcal{V}$ with cardinality K.
compute matrix P from G.
$\mathcal{S}_0 := \emptyset$
$F^{\mathcal{S}_0} := (I - P^{\mathcal{S}_0})^{-1}$
$s = \arg\max \mathbf{1}' \breve{F}^{\emptyset}$, and $\mathcal{S}_1 = \mathcal{S}_0 \cup \{s\}$
$v^{\mathcal{S}_1} = \breve{F}^{\mathcal{S}_0}(:, s)$
for $k = 1$ to $K - 1$ **do**
  $\forall i, j \in \mathcal{I} : F_{ij}^{\mathcal{S}_k \cup \{s\}} = F_{ij}^{\mathcal{S}_k} - \frac{F_{is}^{\mathcal{S}_k} F_{sj}^{\mathcal{S}_k}}{F_{ss}^{\mathcal{S}_k}}$
  $s = \arg\max(\mathbf{1} - v^{\mathcal{S}_k})' \otimes \mathbf{1}' \breve{F}^{\mathcal{S}_k}$, and $\mathcal{S}_{k+1} = \mathcal{S}_k \cup \{s\}$
  $v^{\mathcal{S}_{k+1}} = v^{\mathcal{S}_k} + (\mathbf{1} - v^{\mathcal{S}_k}(s)) \breve{F}^{\mathcal{S}_k}(:, s)$
end for

4.6 Discussion

In this section, we present the comprehensive view of HC model and elaborate its (unifying) relation to the other models by providing multiple interpretations.

Social interpretation. HC can be simply extended to model many real cases that the other influence models fail to cover. As briefly mentioned in Section 4.3, the original HC (4.1), models both “social" and “non-social" influences which cover the observations from the real datasets [25]. The extension of HC which is more flexible in modeling real

Table 4.1: Specifying the equal heat system for existing non-progressive influence models.

Model	Non-Social influence	Weighted edges	Boundary		Init. Cond.		Physical Heat Conduction System
			High $T=1$	Low $T<1$	$=0$	$\neq 0$	
NLT1	√	√		√	√		Circular ring with a fixed-temp.
NLT2	√	√	√	√	√		A rod with two ends, one high one low, and fixed-temp
NLT3		√			√		(Isolated) circular ring
NLT4		√	√		√		Circular ring with a fixed-temp.
Voter						√	(Isolated) circular ring
GLT	√	√		√			Circular ring with a fixed-temp

world cascades is as follows:

$$u(i,t+1) = m\alpha_i + r\gamma_i + (1-\gamma_i-\alpha_i)\sum_{j\in\mathcal{N}(i)} \omega_{ij}u(j,t), \tag{4.12}$$

where, $\sum_{j\in\mathcal{N}(i)}\omega_{ij} = 1$, $\gamma_i, \alpha_i \in [0,1]$, $m = 1$, and $r = 0$. Factor r models the "discouraging" factor like intrinsic *reluctance* of customers toward a new product, and m represents "encouraging" factor like *media* that promotes the new product. These two factors can explain cases where all neighbors of a node are active but the node remains inactive, or when a node becomes active while none of her neighbors are active [25]. Note that all of the formulas and results stated so far is simply applicable to the general HC model (4.12).

Unification of existing non-progressive models. HC (4.1) unifies and extends many of the existing non-progressive models. In the Voter model, a node updates its choice at each time step by picking one of its neighbors randomly and adopting its choice. In other words, the choice rule of node i is the ratio of the number of her active neighbors to her total number of neighbors. Thus, Voter's choice rule is the simplified form of HC's choice rule (4.1) where ω_{ij} is equal to $\frac{1}{d_i}$ (d_i is the out-degree of node i) and all β_is are set to zero. Also, note that having $\beta_i = 0$ indicates that the Voter does not cover the "non-social" influence.

In the *non-progressive* LT (NLT) [74], each node is assigned a random threshold θ

at each time step and becomes active if the weighted number of its active neighbors (at previous time step) becomes larger than its threshold: $\sum_{j\in\mathcal{N}(i)}\omega_{ij}\delta_j(t)\geq\theta_i(t+1)$, where the edge weights satisfy $\sum_{j\in\mathcal{N}(i)}\omega_{ij}\leq 1$. Thus, the choice rule of node i at time $(t+1)$ under the NLT is obtained from the following equation:

$$\begin{aligned} Pr\big(\delta_i(t+1)=1|\mathcal{N}(i)\big) &= Pr\big(\theta_i(t+1)\leq\Sigma\omega_{ij}^{\text{NLT}}\delta_j(t)\big) \\ &= \Sigma\omega_{ij}^{\text{NLT}}\delta_j(t), \end{aligned} \tag{4.13}$$

where the second equality is the result of sampling $\theta_i(t+1)$ from the *uniform distribution* $U(0,1)$. Equation (4.13) is the simplified form of HC's choice rule (4.1), where $b=0$ and $(1-\beta_i)\omega_{ij}^{\text{HC}}=\omega_{ij}^{\text{NLT}}$. Note that since in the NLT b accepts only zero value, this influence model also cannot cover *encouraging* "non-social" influence. Moreover, if the edge weights' gap in NLT, i.e. $g_i=1-\sum_{j\in\mathcal{N}(i)}\omega_{ij}^{\text{NLT}}$, is zero for all the nodes, it cannot model the "non-social" influence at all, since the corresponding β_i's in (4.1) would be equal to zero in that case.

Generalized linear threshold (GLT) is another non-progressive model proposed in [106] to model the adoption process of *multiple* products. Assigning a color $c\in\mathcal{C}$ to each product, a node updates its color, at each time step, by randomly picking one of its neighbors based on its edge weights and adopts the selected neighbor's color. For binary case $|\mathcal{C}|=2$, where we only distinct between adoption of a desired product (active) and the rest of products (inactive), GLT's choice rule reduces to the following equation: $Pr\big(\delta_i(t+1)=1|\mathcal{N}(i)\big)=\frac{\beta}{2}+(1-\beta)\sum_{j\in\mathcal{N}(i)}\omega_{ij}\delta_j(t)$. It is easy to see that this is the restricted form of HC's choice rule (4.1), where nodes are all connected to the bias node with equal weight of β and bias value b has to be $\frac{\beta}{2}$.

Physical interpretation. We showed that the existing non-progressive models are special cases of HC, and in this part we describe their equal heat conduction system which are uniquely specified by the initial and boundary conditions. Table 4.1 summarizes the heat interpretation of the influence models. We introduce four variants of non-progressive LT, based on two factors: seed and gap g_i. NLT1 and NLT2 support non-zero gaps, and NLT2 and NLT4 allows seeds, i.e. nodes in the network that always remain active. The non-progressive LT model presented in [74] is equivalent to NLT2. Reluctance factor and seeds in all models are equivalent to the low and high temperature boundaries respectively, and initial condition addresses the interiors' initial values (z in (4.5)). The

non-social influence and edge weights factors appear in the Laplacian matrix calculation of (4.5). The equivalent physical heat conduction systems are easy to understand, here we just briefly point out the equivalence of the Voter model and the isolated circular ring. Circular ring is a rod whose ends are connected to each other and do not have any energy exchange with outside [69] which explains why the Voter conserves the total initial heat energy, and reaches to an equilibrium with an equal temperature for all of the nodes, i.e., consensus.

Random walk interpretation. Beside the heat conduction view, the random walk prospect helps to gain a better understanding of the models and their relations. Assume that active and inactive nodes are colored black and white respectively. Consider the *original view* of any influence model which is the actual process that unfolds in time, so we look at the time-forward direction. We take a snapshot of the colored network at each time step t. Putting together the sequence of snapshots, the result is a random walk in the "colored graphs" state space with 2^n states. On the other hand, the *dual view* looks at the time-reverse direction of influence models. It is known for both IC-based models (like IC [74] and ConTinEst [43]) and LT-based models (Table 4.1 as well as HC and LT) that a single node from $\mathcal{N}(i)$ is responsible for i's color switch, which we name it as the parent of i. Now assuming that the process has advanced up to the time t, we reverse the process by starting from each node i and follow its ancestors. Here is the point where IC and LT based models separate from each other: due to $\sum_{j\in\mathcal{N}(i)} \omega_{ij} \leq 1$ constraint, ancestors of i in the LT-based models form a random walk starting from node i, which is not the case in IC-based models. Note that we have n random walks that can meet and merge, thus they are known as *coalescing random walks* [8]. This view also helps us to demonstrate the essential difference between progressive and non-progressive models. Dual view of progressive LT model is a *coalescing self-avoiding walks* which is the outcome of randomizing the threshold θ only once at the beginning of the process for the nodes in each realization. This bounds the number of "live" edges [74] connected to each node by one which prevents the creation of "loop" in the influence paths. Note that both counting and finding the probability of self-avoiding walks are $\#P$ hard [31].

Table 4.2: List of networks used in experiments.

		$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{E}\rvert$	Params
Synthetic Networks	Random	1024	-	$[0.5, 0.5; 0.5, 0.5]$
	Hier.	1024	-	$[0.9, 0.1; 0.1; 0.9]$
	Core.	1024	-	$[0.9, 0.5; 0.5, 0.3]$
	ForestFire	1K-300K	$2.5\lvert\mathcal{V}\rvert$	$[0.35, 0.25]$
Real Networks	KClub	34	501	-
	PBlogs	1490	19087	-
	WikiVote	7115	103689	-
	MLWFW	10604	168918	-

4.7 Experiments

In this section, we examine several aspects of C2Greedy and compare it with state-of-the-art methods. Experiments mainly focus on influence maximization and timing aspects. Finally, we present one example of real non-progressive data and illustrate the result of C2Greedy.

4.7.1 Dataset

Table 4.2 summarizes the statistics of the networks that we use throughout the experiments. We work with both synthetic and real networks which we briefly discuss next.

Synthetic network generation. We consider the following types of Kronecker network for extensive performance comparison of our method with the state-of-the-art methods: random [46] (parameter matrix $[0.5, 0.5; 0.5, 0.5]$), hierarchical [33] ($[0.9, 0.1; 0.1; 0.9]$), and core-periphery [85] ($[0.9, 0.5; 0.5, 0.3]$). We generate 10 samples from each network and report the average performance of each method. Edge weights are drawn uniformly at random from $[0, 1]$ and weights of each node's outgoing edges is normalized to 1. For timing experiment, we use ForestFire [33] (Scale-free) network with forward and backward burning probability of 0.35 and 0.25, respectively, and set the outgoing edge weights of node i to $1/|\mathcal{N}(i)|$. The expected density, i.e., number of edges per node, for the resulted ForestFire networks is 2.5.

Real Networks. Zachary's karate club network (KClub) is a small friendship network with 34 nodes and 501 edges [139]. The political blogs network (PBlogs) [2], is a moderate size directed network of hyperlinks between weblogs on US politics with 1490 nodes and 19087 edges. Wikipedia vote network (WikiVote), is the network of who-vote-whom from wikipedia administrator elections [87] with 7115 nodes and 103689 edges. Finally, MLWFW is the network of who-follow-whom in the machine learning research

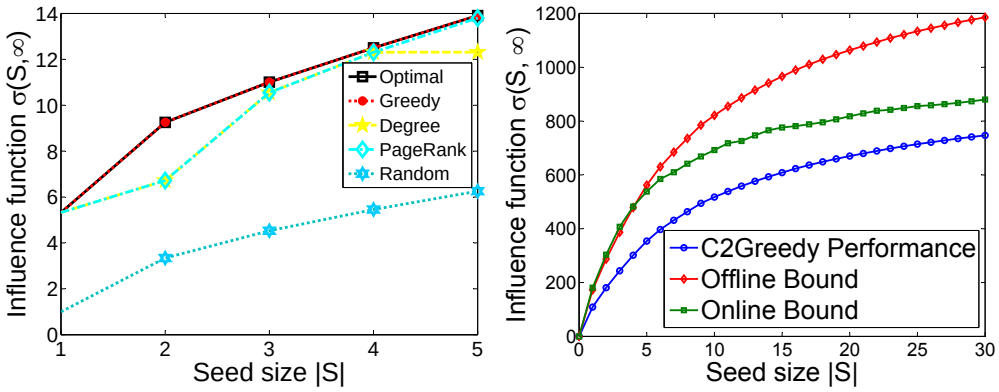
community which we extract from citation networks of combined ACM and DBLP citation network which is available as a part of ArnetMiner [125]. For more information about MLWFW refer to Section 4.7.4.

For all synthetic and real networks, after constructing the network, we add the bias node to the network and connect all nodes to it with weight $\beta_i = 0.1$ and re-normalize the weight of the other edges accordingly.

4.7.2 Influence Maximization

In this section we investigate the performance of C2Greedy in the main task of influence maximization i.e., solving the set function optimization (4.9). Since finding the optimal solution for (4.9) is NP-hard, we compare C2Greedy with optimal solution only for a small network, then for a large network we show that C2Greedy result is close to the online bound [89]. We also compare the performance of C2Greedy with the state-of-the-art methods proposed for solving (4.9) under different (mostly progressive) influence models.

C2Greedy vs. optimal. For testing the quality of C2Greedy method, we compare its performance with the best seed set (determined by brute force) on a small size network. We work with the KClub network for the brute-force experiment with $K = 5$. As Figure 4.1a shows C2Greedy selects nodes that match the performance of the optimal seed set. In the next step, on a larger network, we show that the performance of C2Greedy is close to the known online upper bound [89]. We compute the online and offline bounds of greedy influence maximization [89] with $K = 30$ for PBlogs network. Figure 4.1b illustrates that C2Greedy result is close to the online bound and therefore close to the optimal solution's performance.



(a) C2Greedy vs. optimal.

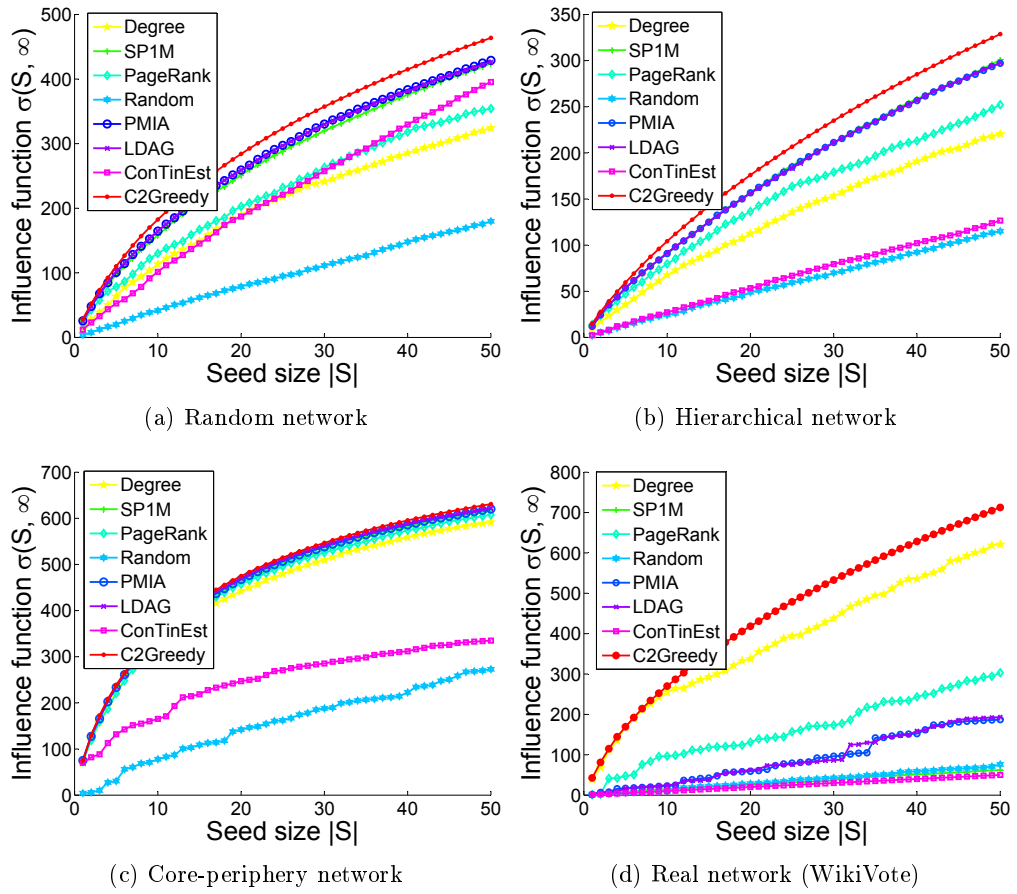
(b) Online and offline bounds.

Figure 4.1: For small network (a) shows C2Greedy matches the optimal performance. For a larger network (b) compares performance of C2Greedy with online and offline bounds.

C2Greedy vs. state-of-the-art. Next, we compare C2Greedy with the state-of-the-art methods of influence maximization over three aforementioned synthetic networks and WikiVote real network. Among baseline methods PMIA [30] and LDAG [31] are approximation for IC and LT models respectively and SP1M [77] is a shortest-path based heuristic algorithm for influence maximization under IC. ConTinEst [64] is a recent method for solving continuous time model of [63] and PageRank is the well-known information retrieval algorithm [20]. Finally, Degree selects the nodes with highest degree as the most influential and Random picks the seed set randomly.

The comparison results are depicted in Figure 4.2. Interestingly, our algorithm outperformed all of the baselines. Strangely, ConTinEst performs close to Random (except in the random network). A closer look at the results for three synthetic networks reveal that except ConTinEst's odd behavior all other methods have persistence rank in performance. C2Greedy is the best method and is followed by PMIA and LDAG, both in second place, which are closely followed by SP1M. PageRank, Degree and Random are next methods in order. In WikiVote real network of Figure 4.2d surprisingly most of the state-of-the-art methods perform terribly poor and Degree (as the KMIP solution to Voter model) is the only competitor of C2Greedy. Result of experiment with WikiVote shows that most influential nodes in a progressive models are not necessary influential in non-progressive ones, and designing non-progressive-specific algorithms (like

C2GREEDY) is required for influence maximization under non-progressive models.



(a) Random network

(b) Hierarchical network

(c) Core-periphery network

(d) Real network (WikiVote)

Figure 4.2: Comparing performance of C2Greedy with state-of-the-art influence maximization methods. Networks of (a), (b), and (c) are synthetic and (d) is a real network.

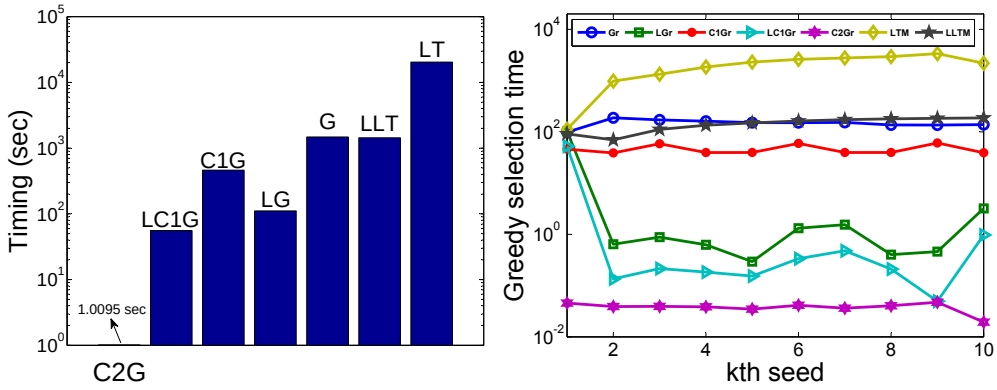
4.7.3 Speed and Scalability

In this part we illustrate the speed benefits of having two closed form updates in the greedy algorithm and also deal with the required single inverse computation of C2GREEDY to prove the scalability of our method.

Closed form benefits. As discussed in Section 4.5, our main algorithm C2GREEDY benefits from closed form computation for both influence spread (4.8) and greedy selection (4.11). To show the gain of these closed form solutions, we run the greedy algorithm

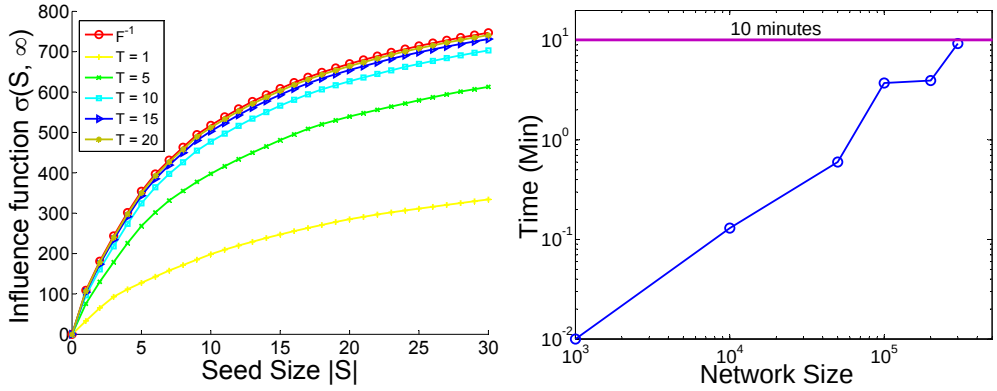
in three different settings. First without using any of (4.8) and (4.11) which we call GREEDY and uses Monte Carlo simulation to estimate the influence spread. Second we only use (4.8) to have the closed form for influence spread without closed form greedy update of (4.11) which results in C1GREEDY, and finally C2GREEDY which uses both (4.8) and (4.11). Note that we can add lazy update of [89] (see Supplementary) to GREEDY and C1GREEDY to get LGREEDY and LC1GREEDY respectively. Finally we include the original greedy method [74] of solving LT model (progressive version of our model) and its lazy variant, with 100 iteration of Monte Carlo simulation. Note that for having a good approximation of influence spread in LT model, simulations are run for several thousand iterations, but here we just want to illustrate that the greedy algorithm for HC is much faster than LT, for which 100 iterations is enough. Figure 4.3a illustrates the speed in log-scale of all seven algorithms for $K = 10$ over the Pblogs dataset [2]. Note that the required time of inverse computation (4.7) is also included. The results confirm that both closed forms decrease the timing *significantly* (1 sec vs. 461 sec for the next best variation) and help the greedy algorithm far more than the lazy update.

Per-seed selection time. The major computational bottleneck of our algorithm is the inverse computation of (4.7). But fortunately this is needed once and at the beginning of the process. Here assuming offline inverse computation, we are interested in the cost of adding each seed. Figure 4.3b compares the cost of selecting k-th seed for the five variation of our algorithm, plus LT and LazyLT all described previously. As expected C2GREEDY requires the lowest computation time per seed. Also, the timing per seed for C2GREEDY is strictly decreasing over the size of $\mathcal{S}$, because the matrix N shrinks, while per seed selection time of LT is increasing on average, because more seeds probably lead to bigger cascades.



(a) Total time (b) Time per seed (sec)

Figure 4.3: In (a) we compare the total timing of seven algorithms to investigate the effect of closed updates on speed and in (b) we show the per-seed required time for the same experiment.



(a) Inverse Approximation (b) Scalability for $K = 10$

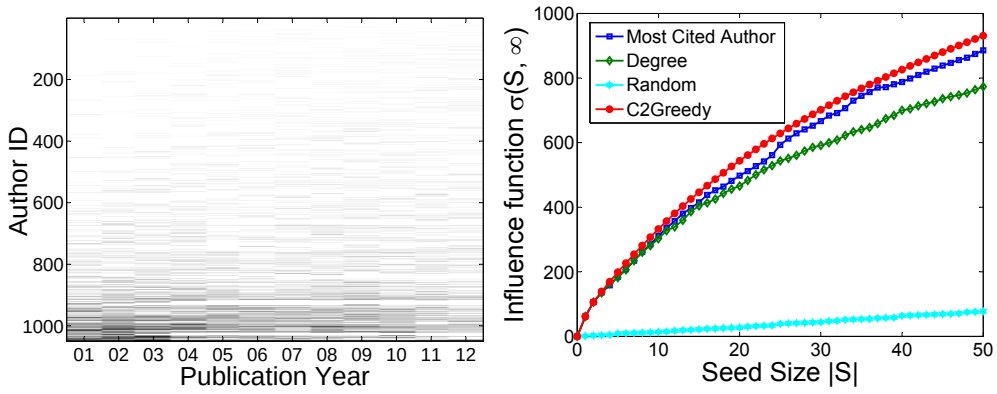
Figure 4.4: Timing for inf. max. in large scale networks by exploiting (a) inverse approximation and (b) parallel programming. Results of (b) are on FF networks with edge density 2.5.

Inverse approximation. Going beyond networks of size 10^4 makes the inverse computation problematic, but fortunately we have a good approximation of the inverse through the following expansion: $F = (I - R)^{-1} \approx I + R^1 + R^2 + ... + R^T$. Since all eigenvalues of R are less than or equal to 1 contribution of $(R)^i$ to the summation drops very fast as i increases. The question is how many terms of the expansion, T,

is enough for our application. Heuristically we choose the (effective) diameter of the graph as the number that provides us with a good approximation of F^{-1}. Note that the ith term of the expansion pertains to the shortest paths of size i between any pair of nodes. Since the graph diameter is the longest shortest path between any pair of nodes, having that many terms gives us a good approximation of F^{-1}. This is also demonstrated by the experimental result of Figure 4.4a where we compare the result of the influence maximization on the WikiVote network with diameter 15, with actual F^{-1} and its approximation for different T's. As discussed when T reaches to the diameter, the result of the algorithm that uses inverse approximation coincides with the algorithm that uses the exact inverse.

Scalability. Finally to show the scalability of C2Greedy we perform influence maximization on networks with sizes up to 3×10^5. For speeding up the large scale matrix computation of the Algorithm 1 we developed an MPI version of our code which allows us to run C2Greedy on computing clusters. Figure 4.4b shows the running time of C2Greedy for ForestFire networks of sizes varying between 1K to 300K with edge density 2.5 (i.e. ratio of edges to nodes) and effective diameter of 10. The MPI code was run on up to 400 cores of 2.8 GHz. As Figure 4.4b indicates even for the largest tested network with 0.3 million nodes and 0.75 million edges C2Greedy takes less than 10 minutes for $K = 10$.

To give a sense of our achievement in scalability we briefly mention the result of one of the state-of-the-art methods: The scalable ConTinEst [43] runs with 192 cores for almost 60 minutes on ForestFire network of size 100K and edge density of 1.5 to select 10 seeds, where our C2Greedy finishes in less than 2 minutes for the similar ForestFire network (100K nodes and density 1.8) with 200 cores.



(a) Non-progressive cascade of ML research topic.

(b) Inf. max. on inferred WFW network, ML-WFW.

Figure 4.5: In (a) we show the existence of non-progressive cascade of ML research topic where white means all papers of the author is about ML. In (b) we compare C2Greedy result with other baselines such as most cited author.

4.7.4 Real Non-Progressive Cascade

Collaboration and citation networks are two well-known real networks that have been studied in social network analysis literature [74, 124]. Here we introduce a new network that represents who-follows-whom (WFW) in a research community. Note that the nodes in the collaboration and citation networks are authors and papers respectively but in WFW network nodes are authors and edges are inferred from citations. A directed edges (u, v) means that author u has cited one of the papers of author v which reveals that u follows/reads papers of v. Here we investigate the "research topic adoption" cascade. Researchers adopt new research topics during their careers and influence their peers along different research communities. The process starts with an arbitrary research topic for each author and they are influenced by the research topic of those they follow and switch to another topic. For example a data mining researcher that follows mostly the papers of machine learning authors is probably going to switch his research topic to machine learning.

For illustration, we consider only the authors who have published papers in Machine Learning (ML) conferences and journals in a given time period. For the list of ML related

conferences and journal we use resources of ArnetMiner project [125]. We consider each time step a year and study the years 2001 - 2012. An author is an *active* ML author in a given year if at least half of his publications in that year was published in ML venues. Figure 4.5a shows the change in the percentage of ML publication of ML authors who has more than 70 publication in years between 2001 and 2012. As Figure 4.5a suggests, cascade of ML research topic is a non-progressive process and researcher switch back and forth between ML and other alternatives. Among 1049 authors of Figure 4.5a about 400 of them are core ML authors who have rarely published in any other topic, but the non-progressive nature of the process is more visible in the rest (bottom part of Figure 4.5a).

Next we perform influence maximization on the inferred WFW network which we call MLWFW network. We extract the MLWFW network from the combined citation network of DBLP and ACM which is publicly available as a part of ArnetMiner project [125] and learn the edge weights similar to the weighted cascade model of [74]. The MLWFW network of 2001 - 2012 time frame consists of 10604 authors and 168918 edges. Figure 4.5b compares the result of influence maximization using C2Greedy and other baselines. Note that other than regular baselines in this specific domain we have another well-known method which is "most cited author" that is equal to selecting authors with highest weighted in-degree in MLWFW network. As Figure 4.5b illustrates, C2Greedy outperforms all of the other methods. Note that the list of K most influential authors in this experiment means that "if" those authors were switching to the ML topic completely (becoming a member of seed set $\mathcal{S}$) they would make the topic vastly popular. Therefore, although the seed set contains the familiar names of well-known ML authors (e.g., Michael I. Jordan and John Lafferty in first and second places), sometimes we encounter exceptions. For example, in the list of top 10 authors selected by C2Greedy we have "Emery N. Brown" who is a renowned neuroscientist with publications in "Neural Computation" journal.

4.8 Proof of Theorem

Proof of Theorem 8. Proof of Theorem 8 is simply an instantiation of Lemma 2 for the case that we add node s as the first seed to the network and get $Q_{is}^{\{s\}} = \frac{F_{is}^{\emptyset}}{F_{ss}^{\emptyset}}$, where $\emptyset$

emphasizes that the bias node is the only boundary. Note that this lemma is general in a sense that absorbing set can contain any type of boundary points, including zero-value node like the bias node and one-value node like a seed node.

□

Proof of Theorem 9. Consider an instance of the NP-complete Vertex Cover problem defined by an undirected and unweighted n-node graph $G = (\mathcal{V}, \mathcal{E})$ and an integer k; we want to know if there is a set $\mathcal{S}$ of k nodes in G so that every edge has at least one endpoint in $\mathcal{S}$. We show that this can be viewed as a special case of the influence maximization (4.9). Given an instance of the Vertex Cover problem involving a graph G, we define a corresponding instance of the influence maximization problem under HC for *infinite time horizon*, by considering the following settings in (4.1): (i) $\omega_{ij} = \omega_{ji} = 1$, if edge $(i - j) \in \mathcal{E}$, otherwise $\omega_{ij} = \omega_{ji} = 0$, (ii) bias node's value is zero $b = 0$, and (iii) β_i for all i's are equal to a known β. Note that since each interior node is connected to the zero-value bias node with edge weight β it cannot have value larger than $1 - \beta$. Hence, if there is a vertex cover $\mathcal{S}$ of size k in G, then one can deterministically make $\sigma(\mathcal{A}, \infty) = k + (n - k)(1 - \beta)$ by targeting the nodes in the set $\mathcal{A} = \mathcal{S}$; conversely, this is the only way to get a set $\mathcal{A}$ with $\sigma(\mathcal{A}, \infty) = k + (n - k)(1 - \beta)$. □

Proof of Theorem 10. As mentioned in Section 4.5.3 when $t \to \infty$ superposition principle applies for HC model. We exploit this fact to prove the submodularity of influence spread. First note that $\sigma(\mathcal{S}, \infty)$ computed from (4.8) is the sum of node values and since the conic combination of submodular functions is also submodular it is enough to show that each node value, i.e., $v(i)$ is submodular to proof Theorem 10. Here we need to work with the general set of bias nodes (compare to single bias node b) which we call ground set $\mathcal{G}$. We introduce a new notation where the value of node i is shown with $v^{\mathcal{S},\mathcal{G}}(i)$. Also seed nodes can have arbitrary value of $\geq b$ instead of all 1 values. For proving the submodularity of $v(i)$ we should prove:

$$v^{\mathcal{T}\cup\{s\},\mathcal{G}}(i) - v^{\mathcal{T},\mathcal{G}}(i) \geq v^{\mathcal{S}\cup\{s\},\mathcal{G}}(i) - v^{\mathcal{S},\mathcal{G}}(i), \mathcal{T} \subseteq \mathcal{S} \tag{4.14}$$

We invoke superposition to perform the subtraction:

$$v^{\{s_{v_L}\},\mathcal{G}\cup\mathcal{T}}(i) \geq v^{\{s_{v_R}\},\mathcal{G}\cup\mathcal{S}}(i), \qquad \mathcal{T} \subseteq \mathcal{S} \tag{4.15}$$

where v_L and v_R emphasize that the value of the new seed node is different in left and right hand side and is qual to $v_L = \big(1 - v^{\mathcal{T},\mathcal{G}}(s)\big)$ and $v_R = \big(1 - v^{\mathcal{S},\mathcal{G}}(s)\big)$. Note that $v_L \geq v_R$ since $\mathcal{T} \subseteq \mathcal{S}$. We can not compare the value of nodes in two different networks unless they share same grounds and seeds with possibly different values for each seed. Therefore, we try to make the grounds of both sides of (4.15) identical by expanding the LHS of (4.15) using superposition law [4]:

$$v^{\{s_{v_L}\},\mathcal{G}\cup\mathcal{T}}(i) = v^{\{s_{v_L}\},\mathcal{G}\cup\mathcal{S}}(i) + v^{\mathcal{D},\mathcal{G}\cup\mathcal{S}\cup s,}(i) \tag{4.16}$$

where $\mathcal{D} = \mathcal{S} - \mathcal{T}$. Although second term of (4.16) is complicated but for our analysis it is enough to note that it is a non-negative number $\alpha \geq 0$. Now the submodularity inequality (4.14) reduces to:

$$v^{\{s_{v_L}\},\mathcal{G}\cup\mathcal{S}}(i) + \alpha \geq v^{\{s_{v_R}\},\mathcal{G}\cup\mathcal{T}}(i) \tag{4.17}$$

Now both sides have the same set of sources and grounds and we now $v_L(u) \geq v_R(u)$ and $\alpha \geq 0$ which completes the proof.

□

Chapter 5

The Effect of Different Couplings on Mitigating Failure Cascade in Interdependent Networks

5.1 Introduction

We now live in an increasingly *connected* world which hinges critically on many *inter-dependent* cyber-physical infrastructure systems. These systems include (smart) power grids, intelligent transportation systems, communication networks and the global Internet. These infrastructures rely on computer and control systems as well as communication networks to sense, collect, estimate the system state, environment and other information, invoke and execute appropriate computations and control strategies to adjust and adapt to changes in the system state and to actuate the physical system components to respond to such changes. The cyber system components also serve as a crucial interface between the physical system components and human operators (as well as end users who are ultimate producers/consumers of much of the information, services or goods that the cyber-physical infrastructures provide).

The inter-dependence of critical cyber-physical infrastructure systems is perhaps best exemplified by the relations between power grids and communication networks where power grids rely on communication networks to deliver the state information of the

power system to the control system and relay control back to the power system, while the communication networks depend on the same power grids for the electrical supply. Due to such interdependence, element faults in one network, e.g., crashes of a few switches in the communication network that are used to relay information and control to a smart grid, can induce failures in the other, i.e., the power grid, which would in turn lead to additional failures in the communication network, thereby triggering a cascade of failures in these two *inter-dependent* networks. It has been reported that a number of electrical blackouts, such as the one in Italy on 28 September 2003 [114], have in fact been caused by such inter-dependency induced cascading failures.

We note that the phenomenon of cascading failures can occur in a *single* network. For example, cascading failures occur frequently in a power grid due to the physical nature of the system as failures of transmission lines or power generators can trigger additional node or line failures due to load imbalance or thermal effect. In a communication network, network element (router or link) failures will trigger network control elements to exchange route control messages and re-compute paths to re-route traffic around failed links/nodes; cascading failures may be triggered due to excessive route re-computation overloads at surviving network elements, which lead to further failures. In a *multi-layered* system of *inter-dependent networks*, failures of network elements in one constituent network (also simply referred to as one layer of the multi-layered system) may not only trigger cascades with the same layer, but also trigger failures of network elements in other constituent networks (layers) of the system. Inter-dependencies across the constituent networks of a multi-layered system can induce cascading failures with very different characteristics and dynamics than those occurring within only one layer, often causing wider and more severe damages to the overall system. To assess and enhance the resiliency of a multi-layered systems of inter-dependent networks, it is therefore imperative to understand how inter-dependencies affect cascading failures within and across constituent networks in a multi-layer system.

In this chapter, we propose a theoretical framework for studying cascading failures in an inter-dependent, multi-layer system, where we consider the effects of cascading failures both *within* and *across* different layers. The goal of the study is to investigate how different *couplings* (i.e., inter-dependencies) between network elements across layers

affect the cascading failure dynamics. For simplicity of exposition, we consider a two-layer system with two constituent networks of equal size, and adopt a simple *one-to-one* coupling map across the two layers. Cascading failures within each layers are modelled using the standard *linear threshold* model[1] . We examine how coupling of nodes of different "importance" or "criticality" (as measured by various metrics e.g., by node degree) from the two constituent networks affect the cascading failure dynamics under varying initial failure sizes and cascading thresholds within each layer. We show that under the *one-to-one* coupling map, that how nodes from two inter-dependent networks are coupled together plays a crucial role in the final size of the resulting failure cascades: coupling corresponding nodes from two networks with equal importance (i.e., "high-to-high" coupling) results in smaller failure cascades than other forms of inter-dependence coupling such as "random" or "high-low" coupling. In particular, given a two-layered system with two identical networks, "high-to-high" coupling produces a *mirror* effect in that the coupling exactly mirrors the cascade within each layer and does not produce additional failures than when the two networks are independent.

5.2 Related Work

Due to its increasing importance, resilience of inter-dependent networks has attracted a flurry of interest from a broad and diverse array of research communities. Using a percolation theory-based framework with random graph models, Buldyrev et al [22] demonstrate that interdependent networks can behave very differently from each of their constituents. In their work – and those of many others, the "robustness" of interdependent networks is quantified in terms of asymptotic statistical properties such as the existence of *giant connected components* under random failures. It is well known from the theory of complex networks that (an ensemble of random) power-law networks are more resilient to random node failures, as there is a phase transition in the fraction of random node failures, below which the giant connected component exists with high probability. In [22] Buldyrev et al show that when nodes from two "robust" power-law networks are *randomly coupled* together *one-to-one*, they become more vulnerable to

[1] We remark that our theoretical framework can be applied to (or generalized to) multi-layer systems with more than two networks with more complex coupling functions and cascading failure models.

random failures in the sense that no giant connected component exists with high probability under any fraction of random node failures. In a follow-up work, Parshani et al. [105] show that decreasing the interdependency of the layers, by decoupling some nodes (as are called autonomous) which do not require any resource from the other layers, the failure cascade can be mitigated. In this work, the nodes were picked *randomly* to become autonomous nodes. Schneider et al. [118] suggest a *centrality* based method for picking the autonomous nodes and show how effectively this method reduces the number of required autonomous nodes by a factor of five compared with the random method. In another work, Brummit et al. [21] pursue the Bak–Tang–Wiesenfeld sandpile model [10] to study failure cascades in inter-dependent networks. They show that adding a few interconnections between the layers of the network is beneficial, but it becomes destructive if the number of interconnections are too many. They find the optimal degree of interdependency in which the failure cascade is minimized.

As in the case of robustness of single networks, the aforementioned characterizations of inter-dependent networks based on random graph models/percolation theory provide useful insight into the *general statistical properties of interdependences* over ensembles of *random* graphs/networks. In practice, however, real networks are *deterministic* and *finite*. In particular, *engineered* infrastructure networks such as power-grids and communication networks, are designed to perform certain specific functions, many of which arguably do not follow the "power-law" degree distribution. Furthermore, although the degree of interdependency is important in controlling failure cascades in interdependent networks, it is not always the case to be able to determine the number of autonomous nodes and in some applications this number is given (the resources are limited). In those cases, designing the way that non-autonomous nodes from different layers are coupled together is another effective solution to control and mitigate failure cascades. Rosato et al. [114] conduct a focused study of the inter-dependency between the Italian power grid and Italian communication network, where they demonstrate that line failures in the Italian power grid network can severely affect the Italian communication network even in the case of moderate interconnection of these two networks. In their study, the authors assume that the nodes in the Italian communication network draw power supply from the *geographically* close nodes in the Italian power grid network. In [109] Ranjan and Zhang propose a graph-theoretical *finite network* model for representing inter-dependent

networks and extend the *structural/topological centrality* measure [110] to develop a robustness metric of inter-dependent networks. Using this robustness metric, they show that both the number of coupled nodes from two inter-dependent networks and how they are coupled together can play a critical role in determining the overall robustness of inter-dependent networks. In [100] Nguyen et al study the *Interdependent Power Network Disruptor* (IPND) optimization problem to identify critical nodes in an inter-dependent power network whose removals maximally destroy its functions. Our work differs from these existing studies in that we not only consider the effects of cascading failures both *within* and *across* different layers, but also investigate how different ways of interdependency ("coupling") affect failure cascades in inter-dependent network. We evaluate the results on both real and synthetic networks.

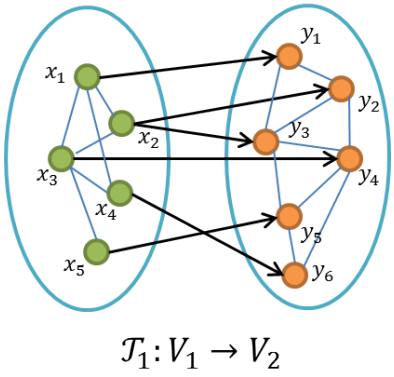


Figure 5.1: Bijective inter-connection of layer 1 to layer 2

5.3 Failure Cascade Model

Consider a network $G(V, E)$, where V is the set of nodes and E is the set of edges. A failure cascade is initiated from a subset of nodes and yields a (larger) set of failed nodes. The failure cascade can be modeled as follows:

$$\mathcal{F} : \mathbb{P}(V) \rightarrow \mathbb{P}(V), \tag{5.1}$$

where $\mathbb{P}(V)$ is the power set of nodes and $\mathcal{F}$ is the failure function in this network. $\mathcal{F}$ depends on the connectivity of the nodes (network topology) and how the failure

cascades through the network. In most real networks, when a node losses a *majority* of its connections to other nodes, the node practically becomes nonfunctional, thus "fails". The linear threshold (LT) model [73] captures this phenomenon in which node i is considered to have failed when the portion of its neighboring nodes $N(i)$ which have failed is larger than some threshold θ:

$$\Sigma_{j \in N(i)} w_{ji} \delta(j) \geq \theta, \tag{5.2}$$

where w_{ij}'s are *importance* weights assigned to the neighboring nodes. In the case of uniform weighting, $w_{ij} = 1$. Considering the LT model as the cascading function, the failure in one network starts from a set of failed nodes and cascades through the network in accordance with eq. (5.2). Note that the failure is considered to be *progressive*, namely when a node fails it does not recover throughout the process [73]. (For the non-progressive LT model, please refer to [57].) In a progressive cascading model, $\mathcal{F}$ is defined deterministically for a fixed θ and a given network G.

Real systems are not always as simple as a single layer network described above. They possess more complex structures, comprised of more than one network (or layer), where nodes in one layer require resources (i.e., power) from nodes in other layers, and in turn supply resources (e.g., control) to nodes in other layers. Such networks, in which the layers are inter-connected to each other, are referred to as *interdependent* networks. In an interdependent network, a node failure in one layer causes its *dependent* nodes in other layers (i.e., those relying on the resources supplied by the failed node to function) also to fail. For example, in Fig. (5.1) if node x_2 fails, its dependent nodes in the other layer, i.e. y_2 and y_3 fail as well. Thus, in interdependent networks an initial failure in one layer may not only cause a failure cascade within the same layer, but also can trigger failure cascades in other layers. The failure cascades in other layers in turn trigger further failures in the original layer, creating a "vicious cycle" which may lead to the break-down of the entire system. While interdependency in such networks is inevitable, it is sometimes possible to carefully "design" the inter-connections between the layers so as to mitigate the effects of failure cascades.

For this purpose, in this work we propose a theoretical framework to model and study failure cascades in interdependent networks. Unlike a single layer network, we argue that in modeling inter-dependent networks, it is important to distinguish the functionality of

"inter-connecting links" (interdependencies) between nodes across layers from the regular links between nodes within a single layer, *as the failure cascading processes within a single failure and across layers are general very different.* For example, failure of a node in general does not automatically leads to the failure of its neighboring nodes *within the same layer* (unless a large portion of neighboring nodes fail under the LT model discussed earlier). On the other hand, failure of a node (i.e., a power supply node) will cause its dependent nodes (e.g., communication or control nodes) in other layers to become non-functional, thus "fail" (with high probability), unless certain protection mechanisms (e.g., backup power) are provisioned. Even in the latter case, such protection mechanisms are often temporal and simply delay the potential failure if the failed nodes are not restored and recovered in time. We present the following general failure cascade model for an interdependent network with two layers $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$, where $\mathcal{F}$ represents the function modeling the failure cascade *within* a layer and $\mathcal{T}$ the function modeling the failure cascade *across* the layers:

$$\begin{aligned} \mathcal{F}_1 &: \mathbb{P}(V_1) \to \mathbb{P}(V_1), \\ \mathcal{F}_2 &: \mathbb{P}(V_2) \to \mathbb{P}(V_2), \\ \mathcal{T}_1 &: V_1 \to \mathbb{P}(V_2), \\ \mathcal{T}_2 &: V_2 \to \mathbb{P}(V_1). \end{aligned} \tag{5.3}$$

Functions $\mathcal{T}_1$ and $\mathcal{T}_2$ are not necessarily injective or surjective. Fig. (5.1) illustrates a bijective function $\mathcal{T}_1$ from layer 1 to layer 2 ($\mathcal{T}_2$ is not shown).

In this work, we show how a proper choice of the functions which model failure cascades *across* the layers can have a significant impact on (triggering/mitigating) the overal failure cascades across the layers. For the ease of exposition, we consider only a bidirectional $\mathcal{T}$ instead of two separate uni-directional $\mathcal{T}_1$ and $\mathcal{T}_2$. In other words, we assume that every node in each layer is served by a unique node (in the other layer) on which it relies for its resources but also for which it supplies the required resources. To have the bijective property in both directions, $\mathcal{T}$ is a "one-to-one" node mapping (or *coupling*) between the two layers, i.e. $\mathcal{T} : V_2 \leftrightarrow V_1$.

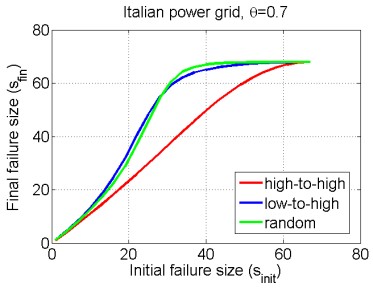


Figure 5.2: Failure cascade in Italian power grid interdependent network for a fixed threshold and three different coupling.

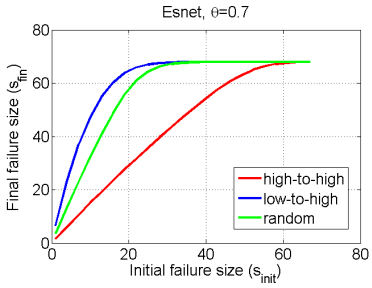


Figure 5.3: Failure cascade in Esnet interdependent network for a fixed threshold and three different coupling.

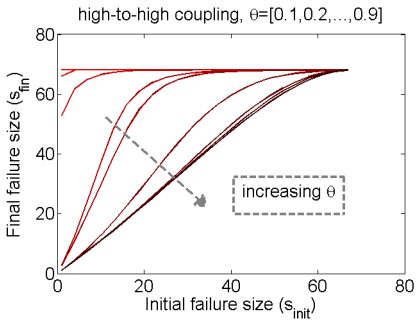


Figure 5.4: Failure cascade in Italian power grid interdependent network for a range of thresholds and high-to-high coupling.

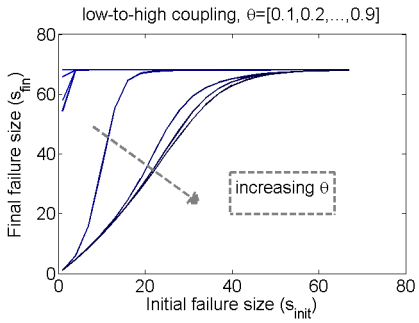


Figure 5.5: Failure cascade in Italian power grid interdependent network for a range of thresholds and low-to-high coupling.

5.4 Experiments and Results

In this section, we investigate the effect of failure cascade modeling functions, i.e. $\mathcal{T}$ and $\mathcal{F}$ in eq. (5.3), on failure cascades across the layers of an interdependent network. Using the LT model as the cascading function within a layer, $\mathcal{F}$ is a function of the threshold θ. For the interdependency ("coupling") function $\mathcal{T}$, we study three representative ways of coupling: 1)"high-to-high" degree coupling, in which the nodes in each layer are sorted based on their degree and are coupled to their corresponding (the same rank) nodes in other layers, 2)"high-to-low" degree coupling with pairing the node in a reverse ordering of their degree, and 3)"random" coupling.

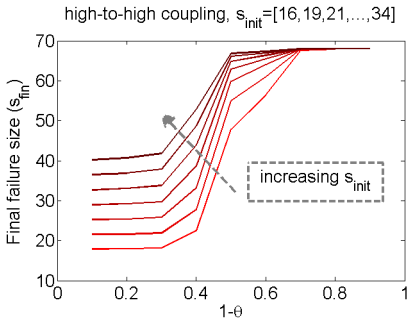


Figure 5.6: Failure cascade in Italian power grid interdependent network for a range of initial failure size and high-to-high coupling

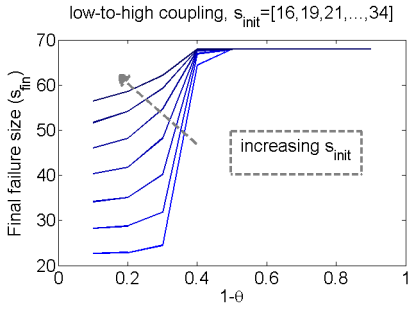


Figure 5.7: Failure cascade in Italian power grid interdependent network for a range of initial failure size and low-to-high coupling.

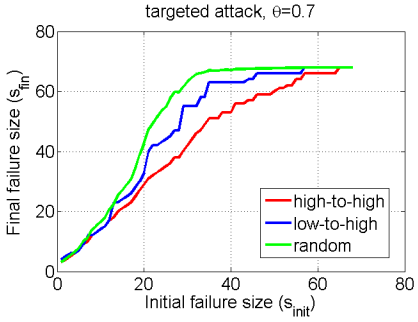


Figure 5.8: Failure cascade in Italian power grid interdependent network for a targeted attack and fixed threshold.

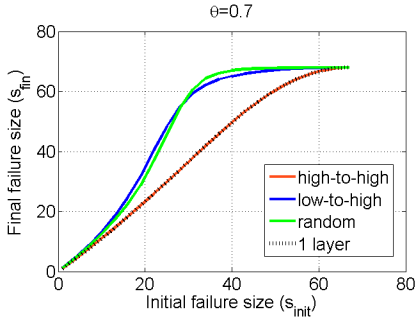


Figure 5.9: Mirroring effect of failure cascade in Italian power grid interdependent network.

We conduct a number of experiments for a wide range of the LT threshold values ($\theta \in [0.1, 0.9]$) and initial failure sizes ($s_{init} \in [1, n]$, n is the number of nodes in each layer). For a fixed size s_{init} of an initial failure, we pick a random s_{init} number of nodes as the initiators of the failure. However, nodes possess different topological importance (centrality), the failure of which can lead to varying sizes of failure cascades (within each layer). Therefor, for each s_{init} we simulate the failure cascades for 10,000 random instance initiators and report the average failure size. Fig. (5.2) shows the results of failure cascades in the Italian power grid network [114] ($n = 68$), when it is coupled with a copy of its own. The experiments are conducted for a fixed threshold of $\theta = 0.7$ in

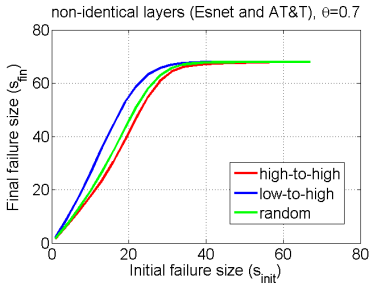


Figure 5.10: Failure cascade in an interdependent network with Italian power grid network as one layer and Esnet as the other.

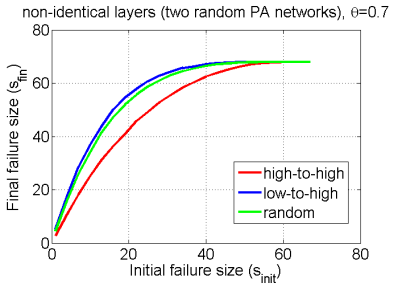


Figure 5.11: Failure cascade in an interdependent network with two layers generated by preferential attachment,

both layers and the results are reported in terms of number of nodes failed in one layer at the end of the cascade process (due to one-to-one coupling, the number of failed nodes are equal in two layers at the end of the cascade). We also perform the exact same set of experiments on the Esnet network, the US DoE energy science network with $n = 68$ number of nodes. The results are reported in fig. (5.3). From figs. (5.2) and (5.3), we see that "high-to-high" coupling show enormously better performance in mitigating the failure cascade than the "low-to-high" and "random" coupling; while "high-to-high" curve is very close to the line $s_{fin} = s_{init}$, two other couplings result in 150% increase in the final failure size over the initial size in some instances for the Italian power grid case (even worse for the Esnet case). The line $s_{fin} = s_{init}$ (not shown in the figures) represents the case where the failure does not cascade and the final failure size is equal to the initial failure size. We also present further failure results for a range of thresholds $\theta \in [0.1, 0.9]$ for the Italian power grid interdependent network in figs. (5.4) and (5.5) for the cases of "high-to-high" and "low-to-high" couplings respectively. Comparing these two figures, it can be inferred that "high-to-high" coupling outperforms "low-to-high" coupling for every θ. Furthermore, increasing θ results in smaller failure cascade sizes, while increasing the initial failure size leads to larger failure cascades. (Due to space limitation, we omit reporting the corresponding results for the case of Esnet, which are very similar.)

Figs. (5.6) and (5.7) reflect the same experiment results explained above, but have been depicted in different way. To avoid making the figures crowded, we have presented

the curves for every three other value of s_{init} from 16 to 34. It can be seen that for every s_{init}, s_{fin} follows a sigmoid-like function in terms of $1-\theta$: there exists one transition point before which the rate of growth is increasing (convex function) and after which the rate of growth is decreasing (concave function). The sigmoid behavior of failure cascades implies that decreasing the threshold up to some transiting point accelerates the failure cascade, but passing that point the rate of cascade slows down. The figures suggest that the transition point is independent of s_{init}; it happens around $\theta \simeq 0.55$ for "high-to-high" coupling and around $\theta \simeq 0.65$ for "low-to-high" coupling. The following general function captures the sigmoid behavior of the final failure size:

$$s_{fin} = \frac{n - g_1(s_{init})}{1 + \exp(-g_2(s_{init})(g_3(\theta)))} + g_1(s_{init}), \tag{5.4}$$

where g_1, g_2, and g_3 are linear functions. For example, for "high-to-high" coupling in the Italian power grid network, these g functions are best fitted with the following linear functions: $g_1(x) = 1.25x - 2$, $g_2(x) = \frac{1}{9}x + \frac{2}{9}$, and $g_3(x) = -10x + 5$. The closed form formulation presented in eq. (5.4) can be useful in predicting the failure size for the large real networks where the simulation is costly or even infeasible in some cases.

Up to now, all the experiments presented in this section have been designed for initial *random* failure. Fig. (5.8) shows the failure result when the initial failure is *targeted*: namely, the failures of more important and central nodes are the results of a targeted attack. In these experiments, the nodes with higher degree are considered to be the initial set of failed nodes. Studies [7] show that the targeted attacks in real networks, where the degree distribution follows a power law distribution, are more harmful than random attacks. Our experiment results show that the "high-to-high" coupling in interdependent network outperforms the other two couplings in targeted attacks as well and assures higher resilience to failure cascades. The failure results obtained for "random" coupling are the average of 10,000 experiments of randomly coupling nodes in the two layers.

As discussed in the previous section, without the interdependency the failure cascade may be minimum in each layer, which is the result of some initiated failure in that layer (i.e., only $\mathcal{F}_1$). Failures in interdependent networks, on the other hand, can cause a "vicious" cycle: when a failure occurs in one layer, besides cascading through the same layer ($\mathcal{F}_1$ in eq. (5.3)), it triggers failures in other layers ($\mathcal{T}_1$); These failures in turn cause further failures in the original layer ($\mathcal{F}_2$ and then $\mathcal{T}_2$) – this cycle continues. To investigate

the effects of different couplings in triggering/mitigating failure cascades, they should be compared against the failure cascade in a single layer network. In fig. (5.9) we compare the failure cascade for the three coupling cases against the failure cascade in one-layer network. This experiment is the same as the experiment in fig. (5.2) but adding the result of the *least possible* failure cascade as well, i.e. failure cascade in one-layer Italian power grid network. It can be seen that, interestingly, the "high-to-high" coupling is in fact equal to having no interdependency at all. This happens due to the *mirroring effect* in which the coupling exactly mirrors the cascade in the two layers and does not lead to further failure than the one is already happening in each of the layers. Thus, leveraging the mirroring effect we are able to design the interdependency functions to minimize the failure cascade in interdependent networks. In the case of identical layers (i.e. layers with the same topology), the best coupling is to pair congruent (equivalent) nodes of the two layers which is the same thing done in "high-to-high" coupling in our experiment fig. (5.9). However, when the layers are not identical, it is more complicated to find the optimum solution. In this case, we should find the best alignment of the layers to benefit from the mirror effect the most possible. We have conducted two experiments on two interdependent networks with non-identical layers: 1) Italian power grid network coupled with Esnet network (fig. (5.10)), and 2) two networks generated by preferential attachment model [11] with the same size of n=68 nodes (fig. (5.11)). The figures indicate that the "high-to-high" coupling outperforms the other two couplings, suggesting that "high-to-high" coupling is more successful in *imitating the mirror effect*, i.e., coupling the congruent nodes of the layers in these experiments.

Part IV

Reachability

Chapter 6

Dynamic Reachability Computations and Pivotality Ranking of Nodes

6.1 Introduction

The reachability information is crucial for a wide range of applications from gene interactions in bioinformatics [129] to XML query processing [26], and for many type of infrastructure networks, be them communication and computer networks, power grids, transportation networks or social networks [137][75][104]. In general, a reachability query $R(s,t)$ has a binary answer with 1 indicating that target node t is reachable from source node s, and 0 representing that it is not. Although several studies have been devoted to devise an efficient and fast algorithm for responding to reachability queries when the network is *static*, fewer solutions are proposed for *dynamic* networks with changes and failures in nodes or edges where a quick recalculation to reachability relationships is needed. Dynamic computation of reachabilities is important for various applications where the network is prone to changes. For example, quick recalculation of reachabilities is required for programming languages garbage collection to balance the reclamation of memory, which might be reallocated, with the performance concerns of the running application [94]. In this chapter, we show that the fundamental matrix of extended network

G^o provides the reachability information of G; more importantly, it answers very efficiently to more advanced reachability queries of $R(s, t, \sim \mathcal{F})$ for the cases that network is changing by a set of failures $\mathcal{F}$ happening in the network, where the size of failures is in $O(1)$ compared to the size of network $|V|$. Our method provides a solution with no update time requirement and $O(1)$ query time, in contrast to state-of-the-art which all need for an update whenever a change (deletion and insertion) happens in the network. Founded on the notion of reachability, we also extend the definition of articulation points to the *directed networks* which is originally defined for undirected networks. We also provide a formulation to compute the articulation points of a network and show that a similar formulation can quantify the load balancing over nodes of a network. Load balancing is important for network robustness against targeted attacks. Through extensive experiments, we evaluate the load balancing in several specific-shaped networks and real-world networks.

More often than not, however, it is not sufficient simply to know that a node s can reach another node t in the network. Additional information is associated with reachability such as how long (e.g., in terms of number of intermediate nodes to be traversed or some other measures of time or cost) or how many possible ways (e.g., in terms of paths) for node s to reach node t. Such information is essential for selecting paths for packet routing or information/commodity delivery, flow scheduling, power management, traffic control, load balancing and so forth in communication and computer networks, power grids and transportation networks. In this chapter, we analyze another piece of important information associated with reachability – which we call pivotality. Pivotality captures how pivotal a role that a third node k or a subset of nodes S may play in the reachability from node s to node t in a given network by quantifying how many (and how long) paths from s to t go through k or S, and how many do not. We quantify this role by exploiting relationships between the hitting time and transit hitting times and examine how much of detour cost k or S can cause. In particular, we propose the *avoidance-transit hitting time* pivotality metric (ATH). Finally, we use several simulated and real-world network examples to illustrate the advantages and utility of avoidance and transit hitting times, especially in comparison with existing metrics proposed in the literature.

6.2 Related Work

Several studies have been devoted to efficient reachability computations in static networks [32, 93, 71, 131] and a few have addressed the reachability computations in networks with changes and dynamic [113, 38]. The studies on dynamic reachability problem propose an oracle, which if be updated after each change (node or edge deletion and insertion), can answer to reachability queries very fast. In one extreme, the query time can become as low as $O(1)$ at the cost of higher amortized update time $O(n^2)$ [112, 38]. Later studies attempted to lower the update time but their suggested query time is non-constant: [66, 113]. In this work, we propose a dynamic reachability oracle to support the changes in the form of (only) deletion *with no update required* as well as *constant query time* $O(1)$. This method is useful for failure prone networks with frequent reahcability query requirement where a fast investigation of several reachabilities and connectivity of entities after failures are of great importance.

Closely related to our pivotality metric, Ranjan and Zhang [110] introduce the notion of *(forced) detour cost* of a random walker from a source s to a target t with respect to a third node k, which is defined as $\Delta H_s^t(k) := H_s^k + H_k^t - H_s^t$. Namely, the (forced) detour cost is the additional steps incurred when a random walker starts at source node s and is forced to first visit the third node k, and then starts from node k to reach target node t vs. the number of the steps it takes starting at source node s and hitting target node t for the first time. Ranjan and Zhang show [110] that aggregated over all pairs of sources and targets, $\sum_s \sum_t \Delta H_s^t(k) = L_{kk}^+$. Here L_{kk}^+ is the diagonal entries of L^+, the Penrose-Moore pseudo-inverse of the graph Laplacian $L = D - A$, where $A = [a_{ij}]$ is the adjacency matrix of a graph (network) and $D = diag[d_i]$, $d_i = \sum_j a_{ij}$, is the diagonal degree matrix. Based on this (forced) detour cost as well as several other interpretations of the diagonal entries L_{kk}^+ of L^+, Ranjan and Zhang advocate $C^*(k) := 1/L_{kk}^+$ as a new node centrality measure – referred to as the *structural* or *topological centrality*, and demonstrate that $C^*(k) := 1/L_{kk}^+$ indeed better captures the structural/topological roles that node k plays in a network than existing centrality metrics, in particular in terms of their roles in the overall network robustness. Motivated by the results in [110], in this paper we aim to provide a more precise characterization of how pivotal a role a third node k may play in the random walks from a source node s to a target node

t by probabilistically quantifying the number of paths from source s to target t that circumvent node k vs. those that traverse node k that the random walker is likely to take. This leads us to introduce two inter-related metrics, *avoidance* and *transit* hitting times, to measure the *pivotality* of node k in the random walks from source s to target t.

6.3 Fast Computation of Network Reachabilities After Failures

We construct network G^o by adding an external node o to network G and connecting all the other nodes to it (node o has no out-going link). Notice that by this operation, the reachabilities of G remains untouched and can be inferred from the reachabilities in G^o. But the merit of G^o is that it has only one recurrent equivalence class and so $F_{s,t}^{\{o\}}$ exists for any pairs of s and t. Recall that $F_{s,t}^{\{o\}}$ is non-zero iff t be reachable from s, since random walk touches every node that are reachable from it before hitting the target node. Therefor, by computing fundamental matrix $F^{\{o\}}$ once, any reachability query $R(s,t)$ can be answered in constant time and $F^{\{o\}}$ simply performs as a look-up table.

Statement 1. *In the extended network G^o, $F_{s,t}^{\{o\}}$ is non-zero if and only if t is reachable from s in the original network G. Specifically, if set $\mathcal{F}$ of nodes fail and become inaccessible, $F_{s,t}^{\{\mathcal{S}\}}$, where $\mathcal{S} = \mathcal{F} \cup \{o\}$, is non-zero if and only if t is still reachable from s in network G after failures.*

Note that network G does not need to be strongly connected, otherwise all nodes were reachable from each other and reachability queries $R(s,t)$ were meaningless.

For an efficient solution to the advanced reachability query $R(s,t,\sim\mathcal{F})$ with node failures in the network, we leverage from the incremental computation of fundamental matrix in Theorem (1). According to this theorem, matrix $F^{\{\mathcal{S}\}}$ can be easily computed from $F^{\{o\}}$ in $O(|\mathcal{F}|)$ time, where $\mathcal{S} = \mathcal{F} \cup \{o\}$. Thus, $F_{s,t}^{\{\mathcal{S}\}}$ that is needed to answer reachability $R(s,t,\sim\mathcal{F})$ can be computed from $F^{\{o\}}$:

$$F_{s,t}^{\{\mathcal{S}\}} = F_{s,t}^{\{o\}} - F_{s,\mathcal{F}}^{\{o\}}(F_{\mathcal{F},\mathcal{F}}^{\{o\}})^{-1}F_{\mathcal{F},t}^{\{o\}}, \tag{6.1}$$

Note that sub-matrix $(F^{\{o\}}_{\mathcal{F},\mathcal{F}})^{-1}$ in the formulation above is always non-singular since $F^{\{o\}}$ is an inverse M-matrix and hence each of its principal sub-matrix is also an inverse M-matrix. Inverse M-matrix is defined to be a matrix whose inverse is an M-matrix. Hence, if $F^{\{o\}}$ is precomputed once, all the reachability queries, i.e. both regular $R(s,t)$ and advanced $R(s,t,\mathcal{F})$, can be answered in constant time. This method for answering the reachability queries is summarized in Algorithm (2). Function $1_{\{b\}}$ is an indicator function which is equal to 1 if $b = True$ and 0 if $b = False$.

Algorithm 2 ANSWERING TO REACHABILITY QUERY

query: $R(s,t,\sim\mathcal{F})$
input: transition matrix P of the extended network G^o
precomputed oracle: $F^{\{o\}} = (I - P_{\setminus o})^{-1}$
output: answer to reachability queries.
if $\mathcal{F} = \emptyset$ **then**
  $R(s,t) = 1_{\{F^{\{o\}}_{s,t} > 0\}}$
else
  $R(s,t,\sim\mathcal{F}) = 1_{\{F^{\{o\}}_{s,t} - F^{\{o\}}_{s,\mathcal{F}}(F^{\{o\}}_{\mathcal{F},\mathcal{F}})^{-1}F^{\{o\}}_{\mathcal{F},t} > 0\}}$
end if

We remark that the reachability queries after edge failures, or even a mixture of node and edge failures, can also be answered with the same proposed method. The only requirement is to add one node in the center of each edge and split the edge into two edges. The failure of this added node now models the failure of the corresponding edge in the original network.

6.4 Articulation Points in Directed Networks

An articulation point, or a cut vertex, is defined to be a node whose removal increases the number of connected components in an undirected network. As an extension to directed networks, Italiano et al. [70] introduced the strong articulation point which is defined for strongly connected networks and refer to a node whose removal increases the number of strongly connected components. In this section, we extended the notion of articulation point to *general directed networks* and define it as a node whose removal decreases the number of *reachabilities* in the network. For instance, if t is reachable from

s and removing node m makes this impossible, node m is an articulation point for the network. Note that the original definition of articulation point for undirected networks complies nicely with our generalized definition. In this section, we provide a formulation to find the articulation points in a general directed networks. We also suggest a similar formulation to quantify the load balancing over nodes of a network.

Recall that the fundamental matrix represent the expected number of visits from nodes before hitting the target (absorbing) node. For example, entry $F_{s,m}^{\{t\}}$ denotes the expected number of visits from m when starting from s and before hitting t for the first time. Note that $F_{s,m}^{\{t\}}$ is strictly greater than 0 for all m that are reachable from s. We use a modified version of fundamental tensor, which we call normalized fundamental tensor (2.4.1), to find the articulation points:

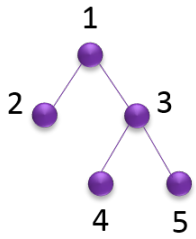
$$\hat{F}_{s,m}^{\{t\}} = \begin{cases} \frac{F_{s,m}^{\{t\}}}{F_{m,m}^{\{t\}}} & \text{if } s, m \neq t \\ 0 & \text{if } s = t \text{ or } m = t, \end{cases} \tag{6.2}$$

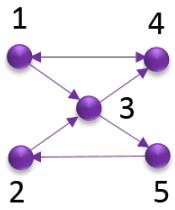
if $F_{s,m}^{\{t\}}$ exists. Normalized fundamental matrix has the following properties: a) has values between 0 and 1: $0 \leq \hat{F}_{s,m}^{\{t\}} \leq 1$, and b) represents the absorption probabilities: $\hat{F}_{s,m}^{\{t\}} = Q_s^{\{m,\bar{t}\}}$. The second property was proved in Theorem (2) and the first one is a direct result of the second property. Recall that $Q_s^{\{m,\bar{t}\}} = 1$ means that m is hit by probability 1 sooner than t when starting from s which implies that m is a gateway for getting to t from s.

Statement 2. *Normalized fundamental tensor captures the articulation points of a network: node m is an articulation point if $\hat{F}_{s,m}^{\{t\}} = 1$ which indicates that node m is located on all paths from s to t. On the other extreme, $\hat{F}_{s,m}^{\{t\}} = 0$ indicates that m is not located on any path from s to t and plays no role for this reachability.*

Figure 6.1 shows two networks, one undirected and one directed, and the corresponding normalized fundamental tensors. The articulation points can be inferred from 1's in the tensor. Note that each node is an articulation point for the reachability of itself to the rest of network and so the diagonal entries are 1. It is interesting to see that counting number of 1's in each column m over the entire tensor represents the number of reachabilities that node m is an articulation point for. The larger the number is, the more critical node m is for network reachabilities. Node 3 in both networks of figure 6.1

is the most critical node by being articulation point for ten and eight reachabilities in the undirected and directed network respectively, ignoring the self-reachbilities.





s \ m	1	2	3	4	5	t
1	0	0	0	0	0	1
2	0	1	0	0	0	1
3	0	0	1	0.5	0.5	1
4	0	0	1	1	0.5	1
5	0	0	1	0.5	1	1

s \ m	1	2	3	4	5	t
1	0	0	0	0	0	1
2	0	1	1	1	0.5	1
3	0	0.5	1	1	0.5	1
4	0	0	0	1	0	1
5	0	1	1	1	1	1

s \ m	1	2	3	4	5	t
1	1	0	0.5	0.33	0.33	2
2	0	0	0	0	0	2
3	1	0	1	0.67	0.67	2
4	1	0	1	1	0.67	2
5	1	0	1	0.67	1	2

s \ m	1	2	3	4	5	t
1	1	0	1	0.75	1	2
2	0	0	0	0	0	2
3	0.5	0	1	0.5	1	2
4	1	0	1	1	1	2
5	0	0	0	0	1	2

s \ m	1	2	3	4	5	t
1	1	0.5	0	0	0	3
2	1	1	0	0	0	3
3	0	0	0	0	0	3
4	0	0	0	1	0	3
5	0	0	0	0	1	3

s \ m	1	2	3	4	5	t
1	1	0	0	0.5	0	3
2	0	1	0	0	0	3
3	0	0	0	0	0	3
4	1	0	0	1	0	3
5	0	1	0	0	1	3

s \ m	1	2	3	4	5	t
1	1	0.67	1	0	0.5	4
2	1	1	1	0	0.5	4
3	0.5	0.33	1	0	0.5	4
4	0	0	0	0	0	4
5	0.5	0.33	1	0	1	4

s \ m	1	2	3	4	5	t
1	1	0.25	0.5	0	0.25	4
2	0	1	1	0	0.5	4
3	0	0.5	1	0	0.5	4
4	0	0	0	0	0	4
5	0	1	1	0	1	4

s \ m	1	2	3	4	5	t
1	1	0.67	1	0.5	0	5
2	1	1	1	0.5	0	5
3	0.5	0.33	1	0.5	0	5
4	0.5	0.33	1	1	0	5
5	0	0	0	0	0	5

s \ m	1	2	3	4	5	t
1	1	0	1	0.75	0	5
2	0.5	1	1	0.5	0	5
3	0.5	0	1	0.5	0	5
4	1	0	1	1	0	5
5	0	0	0	0	0	5

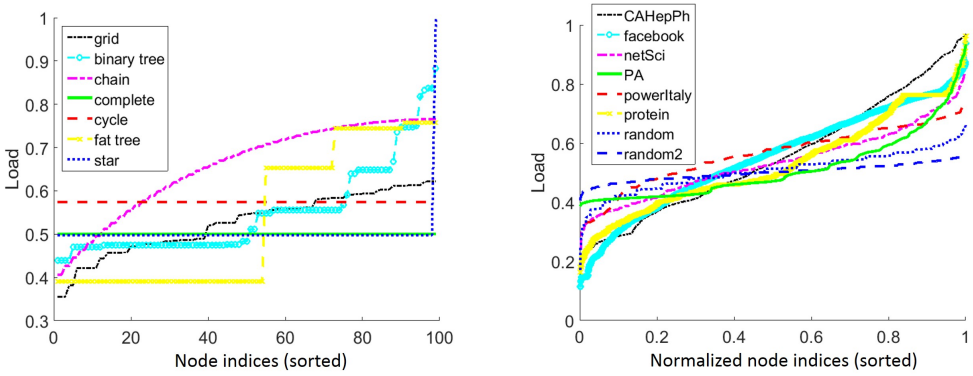
Figure 6.1: Two networks, one undirected and one directed, and the corresponding normalized fundamental tensor

We remark that larger values of $\hat{F}^{\{t\}}_{s,m}$ also implies higher accessibility and/or criticality of node m for the reachability of s to t in the network. The following metric expresses

the overall *load* that node m carries for connectivity of entities in the network:

$$Load(m) = \frac{1}{(n-1)^2} \sum_{s,t} \hat{F}^{\{t\}}_{s,m} \tag{6.3}$$

Calculating the load metric for all nodes in the network, more uniform distribution of loads implies better load balancing. This metric also sheds an insightful light on better understanding of network robustness against *targeted* failures. More balanced networks have higher robustness against the attacks which are targeting a few critical nodes of the network for destructing the connectivity and functionality of the network. Therefore, networks with fairly uniform distributions which are also lower in load value (lower height in y-axis in Figure (6.2)) are the most balanced and reliable networks. Figure (6.2) illustrates the load balancing in specific-shaped networks (a) and real-world networks (b). It can be seen that in the specific-shaped networks with the same number of nodes, complete graph is the most balanced and reliable one followed by cycle graph, where both has completely uniform distribution. It is interesting to see that chain network which lacks only one edge between the first node and the last node compared to cycle graph, experiences a huge difference in the load balancing. Grid has also a flat distribution which implies that load distribution is fairly uniform. In star graph, the central node carries an enormous load which makes the network very vulnerable to targeted attack or failure. It is also interesting to compare the load balancing of 3-ary fat tree [83] against the binary tree. As the real-world networks, the load distribution has been computed for Arxiv High Energy Physics - Phenomenology collaboration network (CAHepPh) [88], Facebook [90], coauthorship network of scientists (netSci) [97], Preferential attachment generative model (PA) [12], Italian power grid [115], protein-protein interaction network [111], and Erdos Renyi random network generative model [45] with two different initial links of 8 (random) and 40 (random2). It can be seen that random2 and random networks followed by Italian power grid have the most uniform load balancing , while PA network shows properties like star network where a few portion of nodes carries the most of the loads. The figure shows that the variance in load size across the nodes is high (skewed load balancing) for Arxiv High Energy Physics - Phenomenology collaboration, Facebook, and Protein-protein interaction networks as well.



(a) (b)

Figure 6.2: Load balancing in a) specific-shaped networks and b) real-world networks

6.5 Node Pivotality in Network Reachability

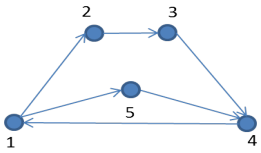


Figure 6.3: Network Example 1

In this section, we examine and quantify how pivotal a role a node k plays in reachability from a source node s to a target node t using our proposed avoidance and transit hitting metrics. In particular, we propose the *avoidance-transit hitting time* pivotality metric (ATH). For a given node k with respect to a pair of source and target nodes s and t, it is defined as follows:

$$e_{ATH}(k) = H_s^{\{t\}} - H_s^{\{t,\check{k}\}} = H_s^{\{t\}} - (H_s^{\{k,\bar{t}\}} + H_k^{\{t\}}). \tag{6.4}$$

Note that if all paths from node s to node t go through a node k^*, then $e_{ATH}(k^*) = 0$. In this case, k^* is the most "pivotal" point of any path from s to t in that all paths rely on k^*. We claim that in such a case, for any other node k, $e_{ATH}(k) \leq 0$; due to space limitation, we will omit the proof here. In general, $e_{ATH}(k)$ can be either positive,

indicating that paths going through node k are overall shorter than an "average" path from node s to node t; or negative, indicating that paths going through node k are overall longer that an "average" path from node s to node t.

For comparison, we also consider other metrics proposed in the literature. We define the *shortest-path* pivotality metric (SHP) to measure the pivotality of node k using the shortest paths only: $e_{SHP}(k) = shp_{st} - (shp_{sk} + shp_{kt})$. The *maximum flow* pivotality metric (MF), $e_{MF}(k)$, measures the amount of the maximum flow from s to t that goes through node k in a flow network, where the weight of edges indicate their capacity. The (classical) *hitting time* pivotality metric (CH) is defined as the negative of the (forced) detour cost defined in [110],

$$e_{CH}(k) := -\Delta H_{s,t}(k) = H_s^{\{t\}} - (H_s^{\{k\}} + H_k^{\{t\}}). \tag{6.5}$$

Notice the similarity between $e_{ATH}(k)$ and $e_{CH}(k)$, except the terms $H_s^{\{k,\bar{t}\}}$ and $H_s^{\{k\}}$. Due to the triangle inequality of the shortest path distance and the hitting time, $e_{SHP}(k) \leq 0$ and $e_{CH}(k) \leq 0$ whereas by definition, $e_{MF}(k) \geq 0$ for all k and all pairs of source and target nodes, s and t. Despite these differences, in terms of ranking of nodes based on their pivotality using each metric, what matters is their relative values: as long as $e(k_1) < e(k_2)$, node k_2 is more "pivotal" than k_1 in terms of reachability from s to t.

6.5.1 Understanding Pivotality Metrics: Examples

Using several simple network examples, in this section we illustrate and compare the behavior of the pivotality metrics defined above. First consider the simple network example shown in Fig. 6.3 where the weight of all edges is 1, i.e., $a_{ij} = 1$. With node 1 being the source and node 4 the target, it is intuitively apparent that node 5 is more "pivotal" than node 2 or node 3, given that it is on the shorter path. The pivotaliy metrics computed using the four methods are shown in Table 6.1. We say that both the MF and CH metrics fail to rank the nodes correctly in that they are not able to recognize the higher pivotality of node 5 over nodes 2 and 3.

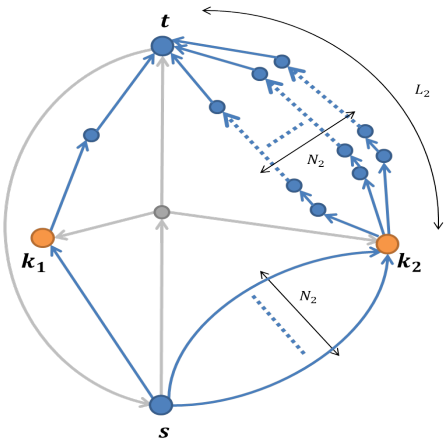


Figure 6.4: Network Example 2

Figure 6.4 provides a more general network example which can help illustrate the different behaviors of the pivotality metrics under study. In this network, there exists a shortest path of length 2 from source s to target t (gray-colored path) interconnected to two groups of (blue-colored) paths passing through k_1 and k_2: a three-hop path from source s via node k_1 to target t, whereas there are N_2 parallel paths going through node k_2, the length of which are L_2+1. If $L_2 = 2$ and $N_2 = 1$ the network is symmetric with respect to k_1 and k_2 and yields equal pivotality for k_1 and k_2 in reachability from s to t (second row of Table 6.2). However, if $N_2 \approx 1$ and $L_2 \gg 2$, intuitively node k_1 plays a more pivotal role than k_2. On the other hand, as the number N_2 of parallel paths going through k_2 increases while their length $L_2 + 1$ is not significantly much longer than 3, say, $L_2 = 3$, node k_2 will play an increasingly more pivotal role in delivering traffic, information or other commodity from node s to node t. Intuitively, there is a trade-off between N_2 and L_2: more parallel paths going through node k_2 will increase its pivotality as it enhances the overall "capacity" from node s to node t; however larger L_2 will diminish its pivotality as longer paths increase the "cost" of using these parallel paths. Despite such intuitions regarding the relative pivotality values of node k_1 and node k_2, if $L_2 > 2$ the SHP pivotality metric will always rank node k_1 higher than k_2 independently of N_2 (for $L_2 = 2$ gives the same ranking to them). Whereas, as long as $N_2 > 1$, the MF pivotality metric will always rank node k_2 higher than node k_1 independently of L_2. Hence both these two metrics fail to capture the differing roles of

Table 6.1: Pivotality metrics in Network Example 1: source node 1 and target node 4

nodes	2	3	5
e_{SHP}	-1	-1	0
e_{MF}	0.5	0.5	0.5
e_{CH}	-3.5	-3.5	-3.5
e_{ATH}	-0.5	-0.5	0.5

Table 6.2: Pivotality metrics (CH and ATH only) in Network Example 2 for various choices of N_2 and L_2

	e_{CH} k_1, k_2	e_{ATH} k_1, k_2
$L_2 = 1, N_2 = 2$	-7,-2.5	-0.75,0.36
$L_2 = 2, N_2 = 1$	-5.14,-5.14	-0.14,-0.14
$L_2 = 2, N_2 = 2$	-8.17,-2.92	-0.17,-0.06
$L_2 = 20, N_2 = 2$	-29.17,-10.42	10.33,-7.56
$L_2 = 20, N_2 = 1$	-15.14,-15.14	7.86,-10.14

node k_2 with varying N_2 and L_2. To evaluate the performance of CH and ATH pivotality metrics in capturing the differing roles of node k_2 with varying N_2 and L_2, some example values are shown in Table 6.2. Based on these results, the CH pivotality metric ranks node k_2 higher than node k_1 as long as $N_2 > 1$, and ranks them the same when $N_2 = 1$ no matter how large is L_2, behaving the same as the MF pivotality metric. However, the ATH pivotality metric ranks successfully node k_1 higher than node k_2 when N_2 is close to 1 and L_2 is quite larger than 2.

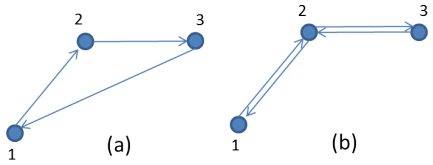


Figure 6.5: Network example 3

The subtle difference in the behaviors of the CH and ATH pivotality metrics lies in the term $H_s^{\{k\}}$ in eq.(6.5) vs. the term $H_s^{\{k,\bar{t}\}}$ in eq.(6.4). Namely, in accounting

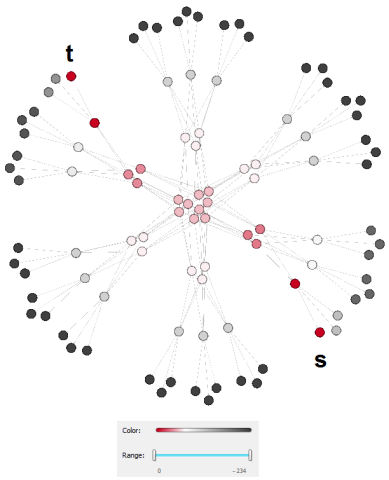


Figure 6.6: Node pivotality ranking in a Fat-tree network for the reachability of the source node s to target node t: red indicates highest pivotality and black shows non-pivotality.

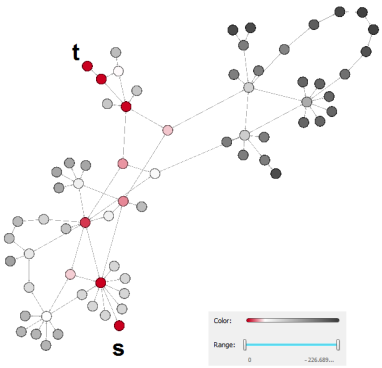


Figure 6.7: Node pivotality ranking in the ESNet network for the reachability of the source node s to target node t: red indicates highest pivotality and black shows non-pivotality.

for the (forced) detour cost, the CH method allows and includes paths/walks from the source node s to the third node k that may have already traversed the target node t; in network example 2, increasing L_2 has a destructive effect on the CH pivotality metric of k_1 by accounting the paths passing through t before hitting k_1, such as the walk $(s-k_2-t-s-k_2-t-...-s-k_1)$, and increasing the term $H_s^{\{k_1\}}$ in eq.(6.5) as the result. In contrast, the ATH method excludes such paths/walks in accounting for the detour cost. As a result, the ATH provides a more precise quantification of the detour cost when a random walker is "forced" to transit a third node k, and thereby how pivotal a role node k plays in the reachability from a source to a target.

The ATH metric allows us to identify nodes that are "superfluous" with respect to the reachability of a source to a target. This can be best illustrated by the two simple examples shown in Fig. 6.5. In both examples, consider node 1 as the source and node 2 as the target. It is obvious that node 3 is "superfluous" with respect to this source-target pair in that node 3 plays no part in the reachability from node 1 to node 2. In other words, if node 3 fails or is removed from the network, the reachability from node 1 to node 2 (and the associated "capacity") is not affected at all. This can be captured by

the fact that in both networks in Figs. 6.5 (a) and (b), the probability of hitting node 3 before node 2 is zero, i.e., $Q_1^{\{3,\bar{2}\}} = 0$. Thus the denominator of the term $H_1^{\{3,\bar{2}\}}$ in eq.(**??**) becomes zero and thus $H_1^{\{3,\bar{2}\}} = \infty$. This renders $e_{ATH}(3) = -\infty$ (see eq.(6.4)), indicating the *non-pivotality* of node 3. In contrast, the CH metric and SHP metric yield $e_{CH}(3) = -3$ and $e_{SHP}(3) = -3$ for Fig. 6.5(a) and $e_{CH}(3) = -4$ and $e_{SHP}(3) = -2$ for Fig. 6.5(b) respectively.

6.5.2 Node Pivotality Ranking using the ATH Metric

Lastly, we apply the node pivotality ranking using our ATH metric to two real-world networks: Fat-Tree [84] and the ESNet [47]. Fat-tree is a special h-ary ($h \geq 2$) "tree-shaped" structure first proposed in [84] for efficient communication with uniform bi-section bandwidth, and for this reason it has been adopted in data center networks [5]. Fig. (6.6) shows 3-ary fat-tree structure with 99 nodes, where the node colors are shaded based on their ATH pivotality measures with respect to the reachability from the source s to the target node t. In the figure, the color spectrum from red to white and then to black shows the range of the ATH value from high to low: the nodes with the larger ATH value, are more pivotal to the reachability from s to t are represented with red and "reddish" colors; in contrast, the nodes that play no part in the reachability from s to t are represented with black color. The results for the ESNet, the DoE energy science network with 68 nodes [47] are shown in Fig. (6.7). Both examples illustrate the efficacy of the ATH metric in correctly capturing and ranking the pivotality of nodes in the reachability from a source node to a target node. Due to space limitation, we do not elaborate on them.

Part V

Routing

Chapter 7

Routing Continuum from Shortest Path to All Path

7.1 Introduction

While in applications like routing in computer networks, the shortest path is the main choice, having alternative paths is beneficial in many cases such as congestion reduction in data networks, avoiding complete predictability of the routing strategy, and increasing the robustness of the network. In wireless networks, using only the "shortest" path is not reliable, because the channels are not stable and their characteristics vary over time. Hence, there is a growing literature on proposing strategies to generate multiple paths and avoid solely relaying on the shortest path [17, 54, 91, 108]. On the other hand, the degeneracy of expected hitting time and failing to measure any notion of distance was shown in [130]. An interpolation or a continuum between expected hitting time and shortest path distance can correct the issue.

In this chapter, we present a novel method for generating a continuum from shortest-path to all-path which is made possible by the concept of random walk avoidance metrics in an evaporation paradigm. By tuning the evaporation parameter α from 0 to 1 a continuum of routing paths from "only the shortest one" to "all possible ones" is yielded. Note that in contrast to the shortest-path scheme that only the shortest path from source node to target node is traversed, in the random walk all-path scheme any path from source node to target node has a non-zero probability to be traversed, and hence

its expected length is equal to classical hitting time. The proposed continuum method, in contrast to previous related work, is not limited to merely generating a continuum of paths or computing the distances, but provides more comprehensive insight and analysis about the network and can be generalized to more complicated cases by addressing the following capabilities all under one framework:

1. Provides a closed form formulation for computing the continuum distances,
2. Provides an efficient routing strategy,
3. Is generalized to support cases with multiple targets,
4. Has a flexible design to generate logical flow instead of stochastic flow,
5. Suggests a novel shortest path method
6. Builds a unifying framework for network measure computations such as centrality measures, distance measures, and topological index.

7.2 Related Work

Generating a continuum from shortest-path to all-path has attracted attentions in recent years. Li et al. [92] proposed a theoretical framework based on mixed (weighted) L_1/L_2-norm optimization as a trade-off between latency and energy dissipation to generate a routing continuum from all-path to shortest-path when a tuning parameter ranges from 0 to large values. For each choice of tuning parameter and source-target pair, this optimization computes the distribution of flows on every edge in the network to determine the edges for the routing purposes. A similar algorithm, called p-resistance, was suggested by Alamgir and Luxburg [6] where a parameter tunes the preference toward L_1 or L_2 norms. Although these two algorithms [92, 6] provide a practical routing strategy based on their continuum generative model, they lack a tractable and closed form formulation for computing all the pairwise *distances*; a separate optimization of order $O(n^3)$ is required for each pair of source-target which makes the algorithm computationally expensive $O(n^5)$. In addition, they cover only undirected networks. On the other hand, there are a class of works which suggest tractable expressions for

computing continuum distances, but no routing strategy is provided. Tahbaz-Salehi and Jadbabaie [123] proposed a continuum over a one-parameter family of algorithms based on Log-Sum-Exp function which converges to Bellman-Ford iterations (shortest-path distance), as one extreme and to mean hitting time iterations (all-path distance), as the other extreme. In another work, logarithmic forest distances proposed by Chebotarev [27] generates a family of distances based on matrix forest theorem. It computes a matrix of distances tuned by parameter α after a sequence of processes. Francoisse et al. [51] form an optimization problem over the path probabilities to minimize the total expected cost subject to a relative entropy constraint. They show that the solution is a Boltzmann distribution over all set of paths and derive a closed form formulation which yields the distances. However, they do not propose any method for finding the paths and the usage portion of each edge in routing purposes. In other words, no routing information and method to pick the edges for different values of tuning parameter are provided in these works.

7.3 Theoretical Framework for Generating the Continuum

We first explain how to form an evaporation paradigm G_α from network G, and then show that a continuum from shortest path to all path over G can be generated by using the avoidance metrics over G_α and tuning α from 0 to 1.

Evaporation paradigm G_α is obtained by multiplying factor $\alpha^{w_{ij}}$ into transition probability P_{ij} of G for all edges $\forall e_{ij} \in E$, where $0 < \alpha < 1$, and adding one (imaginary) node to network, denoted by o, to which every other node i is connected with transition probability $1 - \sum_{j \in \mathcal{N}(i)} \alpha^{w_{ij}} P_{ij}$.

$$P_{ij}(\alpha) = \begin{cases} P_{ij}\alpha^{w_{ij}} & \text{if } i, j \neq o \\ 1 - \sum_{k \in \mathcal{N}(i)} \alpha^{w_{ik}} P_{ik} & \text{if } i \neq o \text{ and } j = o \\ 0 & \text{if } i = o \text{ and } j \neq o \\ 1 & \text{if } i, j = o \end{cases} \tag{7.1}$$

Thus the new transition probability matrix $P(\alpha)$, belonging to G_α, is an $(n+1)\times(n+1)$ row-stochastic matrix whose main principal $n \times n$ submatrix is $P_{11}(\alpha) = P \odot \alpha^W$, where $\odot$ is the element-wise product. Now with the new transition probability matrix

$P(\alpha)$, we compute the avoidance metrics $U_s^{\{t,\bar{o}\}}(\alpha)$ and $F_{sm}^{\{t,\bar{o}\}}(\alpha)$ (from (3.25) and (3.21) respectively), and generate the routing continuum based on the following theorems.

Theorem 11 (Routing Continuum: Path Distances)**.** *Consider weighted network G with at least one path from node s to node t. Varying parameter α from 0 to 1 in the avoidance hitting cost of the corresponding evaporating network G_α yields a continuum from the shortest-path distance to all-path distance (hitting cost distance) from node s to node t in G:*

a) If $\alpha \to 0$, $U_s^{\{t,\bar{o}\}}(\alpha)$ converges to the shortest-path distance from s to t in G,

b) If $\alpha \to 1$, $U_s^{\{t,\bar{o}\}}(\alpha)$ converges to the hitting time distance from s to t in G; More precisely, $U_s^{\{t,\bar{o}\}}(\alpha)$ is exactly equal to the hitting cost distance for $\alpha = 1$,

c) If $\alpha_1 < \alpha_2$, $U_s^{\{t,\bar{o}\}}(\alpha_1) \leq U_s^{\{t,\bar{o}\}}(\alpha_2)$.

The intuition behind Theorem (11) is that decreasing α, the probability of evaporation in paths increases and when α goes to zero, the probability of longer paths become negligible compared to the probability of the shortest path, and only the shortest path survives. In addition, the non-zero entries of matrix $F^{\{t,\bar{o}\}}(\alpha)$ become the indicators of the involved nodes lying on the shortest path when α goes to zero, which is demonstrated in the next theorem.

Theorem 12 (Routing Continuum: Node Flows)**.** *Consider weighted network G with at least one path from node s to node t. For $\alpha \to 0$ in the corresponding evaporating network G_α, the entries of s-th row of the avoidance fundamental matrix, i.e. $F_{sm}^{\{t,\bar{o}\}}(\alpha)$ for $\forall m \in \mathcal{T}$, determine the following information regarding the shortest path from s to t in network G:*

a) If $\lim_{\alpha\to 0} F_{sm}^{\{t,\bar{o}\}}(\alpha) = 0$, no shortest path from s to t passes through node m.

b) If $\lim_{\alpha\to 0} F_{sm}^{\{t,\bar{o}\}}(\alpha) = 1$, node m is located on all of the shortest paths from s to t.

c) If $0 < \lim_{\alpha\to 0} F_{sm}^{\{t,\bar{o}\}}(\alpha) < 1$, a fraction of the shortest paths from s to t pass through node m.

d) As an immediate result of part (c), there exists more than one shortest path from s to t if and only if $\exists m, 0 < \lim_{\alpha\to 0} F_{sm}^{\{t,\bar{o}\}}(\alpha) < 1$.

According to this theorem, computing the s-th row of the avoidance fundamental tensor for $\alpha \to 0$, we can find all of the nodes located on the shortest path(s) from s to

t. In addition, we can compute the shortest path length L_{st} by summing up over this row (3.24). In addition, we can find routing continuum edge probabilities (aka how to choose the next edge in a routing) from matrix $\dot{P}$ based on the following theorem:

Theorem 13 (Avoidance Paradigm to Classical Paradigm Transformation 7)**.** *Network G with avoiding node o and target node t can be transformed to network $\dot{G}$ without node o and the same target t such that the avoidance metrics in the former network turn into the classical metrics in the latter network, i.e. $F_{sm}^{\{t,\bar{o}\}} = \dot{F}_{sm}^{\{t\}}$, $H_s^{\{t,\bar{o}\}} = \dot{H}_s^{\{t\}}$, and $U_s^{\{t,\bar{o}\}} = \dot{U}_s^{\{t\}}$. The transformation function between transition matrix $\dot{P}$ belonging to $\dot{G}$ and P belonging to G is as follows:*

$$\dot{P}_{ij} = P_{ij} \frac{Q_j^{\{t,\bar{o}\}}}{Q_i^{\{t,\bar{o}\}}} \tag{7.2}$$

Corollary 9 (Routing Continuum: Edge Probabilities)**.** *The probabilities assigned to edges for the routing strategy and each choice of α can be obtained from:*

$$\dot{P}_{ij}(\alpha) = P_{ij}\alpha^{w_{ij}} \frac{Q_j^{\{t,\bar{o}\}}(\alpha)}{Q_i^{\{t,\bar{o}\}}(\alpha)} = P_{ij}\alpha^{w_{ij}} \frac{F_{jt}^{\{o\}}(\alpha)}{F_{it}^{\{o\}}(\alpha)}, \tag{7.3}$$

where $Q^{\{t,\bar{o}\}}(\alpha)$ is computed from (2.16) and over evaporation transition probability matrix (7.1). The second equality is resulted from Lemma (2). Algorithm (3) summarizes our method for computing these three metrics to find the continuum information for each choice of α.

Algorithm 3 COMPUTING CONTINUUM PATH INFORMATION FOR EVERY CHOICE OF α

input: Probability transition matrix P, weight matrix W, tuning parameter α

output: Path lengths indicated by $U^{\{t,\bar{o}\}}(\alpha)$, node flows indicated by $F^{\{t,\bar{o}\}}(\alpha)$, and routing edge probabilities indicated by $\dot{P}^{\{t\}}(\alpha)$

initialization: $P_{11}(\alpha) = \alpha^W \odot P$

$F^{\{o\}}(\alpha) = (I - P_{11}(\alpha))^{-1}$

compute $\dot{P}_{ij}^{\{t\}}(\alpha)$ **for every target** t **and every edges** e_{ij}**:** $\dot{P}_{ij}^{\{t\}}(\alpha) = P_{ij}(\alpha)\frac{F_{jt}^{\{o\}}(\alpha)}{F_{it}^{\{o\}}(\alpha)}$

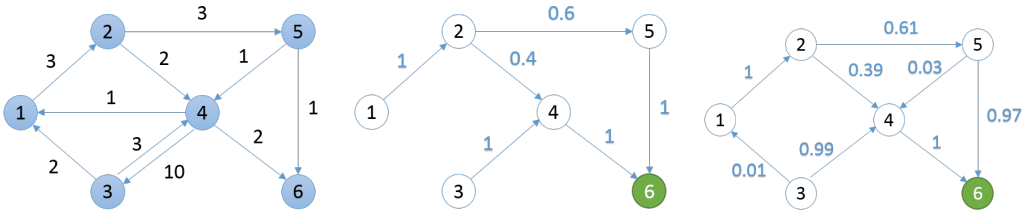
compute $F_{sm}^{\{t,\bar{o}\}}(\alpha)$ **for every triplet** (s,m,t)**,** $s,m \neq t$**:** $F_{sm}^{\{t,\bar{o}\}}(\alpha) = F_{mt}^{\{o\}}(\alpha)(\frac{F_{sm}^{\{o\}}(\alpha)}{F_{st}^{\{o\}}(\alpha)} - \frac{F_{tm}^{\{o\}}(\alpha)}{F_{tt}^{\{o\}}(\alpha)})$

compute $U_s^{\{t,\bar{o}\}}(\alpha)$ **for every pair** (s,t)**,** $s \neq t$**:** $r_m^{\{t,\bar{o}\}}(\alpha) = \sum_{j\in\mathcal{N}(m)} w_{mj}\dot{P}_{mj}^{\{t\}}(\alpha)$, $U_s^{\{t,\bar{o}\}}(\alpha) = \sum_m F_{sm}^{\{t,\bar{o}\}}(\alpha) r_m^{\{t,\bar{o}\}}(\alpha)$

In this algorithm, the equations for computing $\mathrm{P}_{ij}^{\{t\}}$, $F_{sm}^{\{t,\bar{o}\}}$, and $U_s^{\{t,\bar{o}\}}$ come from Eq. (7.3), relation (3.36), and Eq. (3.27) respectively. It can be seen that the worst complexity of computing *all pair-wise paths and distances* for any choice of α is $O(n^3)$.

7.3.1 Network Example

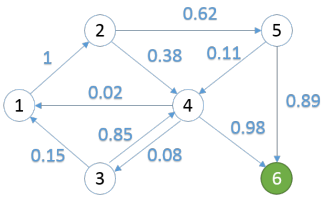
The routing continuum for the network displayed in Fig. (7.1a) is described through main indicators of paths, i.e. $\mathrm{P}^{\{t\}}(\alpha)$, $F^{\{t,\bar{o}\}}(\alpha)$, $U^{\{t,\bar{o}\}}(\alpha)$, which are computed for five different values of α and target $t = 6$, and presented in Table (7.1). The routing strategy in terms of edge probability $\mathrm{P}^{\{t\}}(\alpha)$ for these five different values of α are depicted in Fig. (7.1b-f).



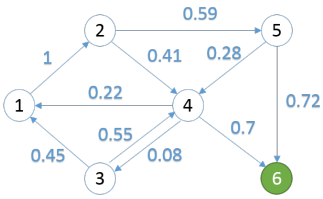
(a) directed weighted network example (weights on the edges)

(b) $\alpha = 0.0001$

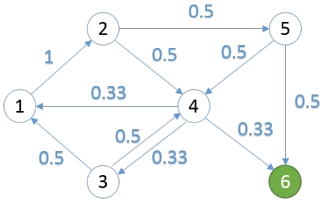
(c) $\alpha = 0.3$



(d) $\alpha = 0.6$



(e) $\alpha = 0.9$



(f) $\alpha = 1$

Figure 7.1: Routing continuum: (b)-(f) show routing edge probabilities for network example in (a) and for different values of α which generate a continuum from shortest path to all path. The weights on the edges in (a) represent the cost of edges and in (b)-(f) indicates the routing edge probabilities. Target is node 6.

α	$\mathrm{P}^{\{6\}}(\alpha)$	$F^{\{6,\bar{o}\}}(\alpha)$	$U^{\{6,\bar{o}\}}(\alpha)$
0.0001	0 1 0 0 0 0 0 0 0 0.4 0.6 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1	1 1 0 0.4 0.6 0 0 1 0 0.4 0.6 0 0 0 1 1 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0	7 4 5 2 1 0
0.3	0 1 0 0 0 0 0 0 0 0.39 0.61 0 0.01 0 0 0.99 0 0 0 0 0 0 0 1 0 0 0 0.03 0 0.97 0 0 0 0 0 1	1 1 0 0.41 0.61 0 0 1 0 0.41 0.61 0 0.01 0.01 1 0.99 0.01 0 0 0 0 1 0 0 0 0 0 0.03 1 0 0 0 0 0 0 0	7.04 4.04 5.04 2.00 1.06 0
0.6	0 1 0 0 0 0 0 0 0 0.38 0.62 0 0.15 0 0 0.85 0 0 0.02 0 0 0 0 0.98 0 0 0 0.11 0 0.89 0 0 0 0 0 1	1.01 1.01 0 0.45 0.63 0 0.01 1.01 0 0.45 0.63 0 0.17 0.17 1.00 0.93 0.10 0 0.02 0.02 0.00 1.01 0.01 0 0 0 0 0.11 1 0 0 0 0 0 0 0	7.19 4.19 5.73 2.13 1.23 0
0.9	0 1 0 0 0 0 0 0 0 0.41 0.59 0 0.45 0 0 0.55 0 0 0.22 0 0.08 0 0 0.71 0 0 0 0.28 0 0.72 0 0 0 0 0 1	1.18 1.18 0.05 0.70 0.70 0 0.18 1.18 0.05 0.70 0.70 0 0.70 0.70 1.08 0.99 0.41 0 0.31 0.31 0.09 1.23 0.18 0 0.09 0.09 0.03 0.34 1.05 0 0 0 0 0 0 0	9.10 6.10 9.44 5.10 2.41 0
1	0 1 0 0 0 0 0 0 0 0.5 0.5 0 0.5 0 0 0.5 0 0 0.33 0 0.33 0 0 0.33 0 0 0 0.5 0 0.5 0 0 0 0 0 1	1.82 1.82 0.55 1.64 0.91 0 0.82 1.82 0.55 1.64 0.91 0 1.45 1.45 1.64 1.91 0.73 0 1.09 1.09 0.73 2.18 0.55 0 0.55 0.55 0.36 1.09 1.27 0 0 0 0 0 0 0	19.36 16.36 21.09 17.82 9.91 0

Table 7.1: Continuum path indicators for target node $t = 6$ and different choices of α for network example in Fig. (7.1a)

$U^{\{6,\bar{o}\}}(\alpha)$ indicates the vector of distances from all nodes to node 6. It can be seen

for α close to zero ($\alpha = 0.0001$ in Table (7.1)) these distances are the same as shortest path distances. For larger α's these distances grow monotonically until $\alpha = 1$ where the classical hitting cost distances are obtained $U^{\{6,\bar{o}\}}(\alpha = 1) = H^{\{6\}}$.

$F^{\{6,\bar{o}\}}(\alpha)$ represents the stochastic flow of nodes to target node 6. It is specially meaningful for two extreme cases of $\alpha = 0.0001$ and $\alpha = 1$; e.g. $F_{1j}^{\{6,\bar{o}\}}(\alpha = 0.0001)$ indicates the stochastic portion of shortest paths from node 1 to 6 that pass through node j, which is 0.4 for $j = 4$, 0.6 for $j = 5$, 1 for $j = 2$ which implies that all of the shortest paths from 1 to 6 pass through node 2, and 0 for $j = 3$ indicating no shortest path from 1 to 6 pass through node 3. Existence of any value larger than 0 and smaller than 1 in i-th row of $F^{\{t,\bar{o}\}}(\alpha \to 0)$ indicate the existence of *multiple* shortest paths from i to t.

For the other extreme $\alpha = 1$, $F^{\{6,\bar{o}\}}(\alpha = 1)$ is representing the expected visit times in regular random walks, i.e. classical fundamental matrix $F^{\{6\}}$.

$\mathrm{P}^{\{6\}}(\alpha)$ is the matrix of edge probabilities for routing purposes. In other words, when a packet arrives at node i it is forwarded over edge e_{ij} with probability $\mathrm{P}_{ij}^{\{6\}}(\alpha)$. Thus $\mathrm{P}_{ij}^{\{6\}}(\alpha)$ indicates the usage portion of edge e_{ij} for routing packets from i to $t = 6$ and for parameter α. For $\alpha = 0.0001$ (shortest path case), it can be seen that edges not belonging to shortest paths have zero probability (and so not shown in the figure), and the non-zero-probability edges form the shortest DAG from all the nodes to target node 6 (Fig. (7.1b)).

7.4 Generalization for Multiple Targets

The advantage of the proposed method is that the routing continuum can be easily extended to a *set* of targets. In terms of random walk, this means that the stopping criteria is the moment that random walk hits the first node in the target set. For the case of $\alpha \to 0$ where the continuum converges to the shortest path, this target generalization in fact picks the shortest one among the set of shortest paths to nodes in the target set, i.e. for target set T the following equations hold $U_s^{\{T,\bar{o}\}}(\alpha \to 0) = \min_{t \in T} U_s^{\{t,\bar{o}\}}(\alpha \to 0) = \min_{t \in T} L_{st}$. Hence, our method automatically reports the minimization result when the objective is to find the shortest path to a set of nodes with no need to do the computations separately for every target. Note that in general,

however, this target generalization does not perform as a minimization operation for larger α's: $U_s^{\{T,\bar{o}\}}(\alpha) \neq \min_{t\in T} U_s^{\{t,\bar{o}\}}(\alpha)$. Algorithm (4) provides the general form of the continuum method presented in Algorithm (3). Fig. (7.2) displays the survived paths for three choices of α and target set $T = \{5, 6\}$, and the corresponding edge probabilities for the routing strategy. α values.

Algorithm 4 COMPUTING CONTINUUM PATH INFORMATION FOR EVERY CHOICE OF α FOR TARGET SET T

input: Probability transition matrix P, weight matrix W, tuning parameter α, target set T

output: Path lengths indicated by $U^{\{T,\bar{o}\}}(\alpha)$, node flows indicated by $F^{\{T,\bar{o}\}}(\alpha)$, and routing edge probabilities indicated by $\mathrm{P}\!\!\!\!\mathrm{P}^{\{T\}}(\alpha)$

initialization: $P_{\mathcal{T}\mathcal{T}}(\alpha) = \alpha^W \odot P$

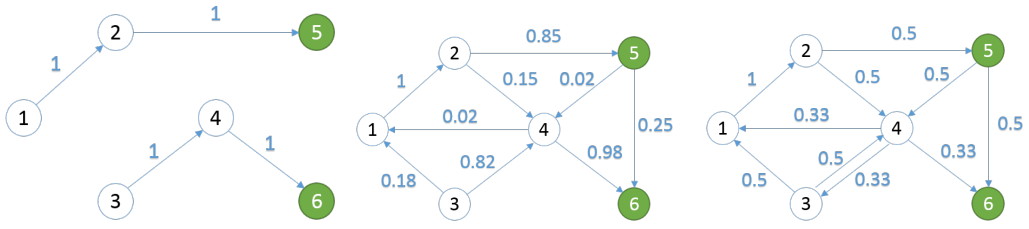
$F^{\{T,o\}}(\alpha) = (I - P_{\mathcal{T}\mathcal{T}}(\alpha))^{-1}$

$Q^{\{T,\bar{o}\}}(\alpha) = F^{\{T,o\}}(\alpha) \sum_j [P_{\mathcal{T}\mathcal{A}}(\alpha)]_{:j}$

compute $\mathrm{P}\!\!\!\!\mathrm{P}_{ij}^{\{T\}}(\alpha)$ **for every edge** e_{ij}**:** $\mathrm{P}\!\!\!\!\mathrm{P}_{ij}^{\{T\}}(\alpha) = P_{ij}(\alpha)\frac{Q_j^{\{T,\bar{o}\}}(\alpha)}{Q_i^{\{T,\bar{o}\}}(\alpha)}$

compute $F_{sm}^{\{T,\bar{o}\}}(\alpha)$ **for every pair** (s, m)**,** $s, m \notin T$**:** $F_{sm}^{\{T,\bar{o}\}}(\alpha) = F_{sm}^{\{T,o\}}(\alpha)\frac{Q_m^{\{T,\bar{o}\}}(\alpha)}{Q_s^{\{T,\bar{o}\}}(\alpha)}$

compute $U_s^{\{T,\bar{o}\}}(\alpha)$ **for every** s**,** $s \notin T$**:** $r_m^{\{T,\bar{o}\}}(\alpha) = \sum_{j\in\mathcal{N}(m)} w_{mj}\mathrm{P}\!\!\!\!\mathrm{P}_{mj}^{\{T\}}(\alpha)$,
$U_s^{\{T,\bar{o}\}}(\alpha) = \sum_m F_{sm}^{\{T,\bar{o}\}}(\alpha) r_m^{\{T,\bar{o}\}}(\alpha)$



(a) $\alpha = 0.0001$ (b) $\alpha = 0.5$ (c) $\alpha = 1$

Figure 7.2: Routing continuum for target set $\{5, 6\}$: routing edge probabilities $\mathrm{P}\!\!\!\!\mathrm{P}^{\{5,6\}}(\alpha)$ for network example in Fig. (7.1a) for three different values of α which generates a continuum from shortest path to all path.

α	$P_s^{\{5,6\}}(\alpha)$	$F^{\{5,6,\bar{o}\}}(\alpha)$	$U^{\{5,6,\bar{o}\}}(\alpha)$
0.0001	0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0	1 1 0 0 0 1 0 0 0 0 1 1 0 0 0 1	6 3 5 2
0.5	0 1 0 0 0 0 0 0 0 0.15 0.85 0 0.18 0 0 0.82 0 0 0.02 0 0 0 0 0.98 0 0 0 0.02 0 0.25 0 0 0 0 0 0	1 1 0 0.15 0 1 0 0.15 0.19 0.19 1 0.85 0.02 0.02 0 1	6.16 3.16 5.64 2.09
1	0 1 0 0 0 0 0 0 0 0.50 0.50 0 0.50 0 0 0.50 0 0 0.33 0 0.33 0 0 0.33 0 0 0 0.50 0 0.50 0 0 0 0 0 0	1.43 1.43 0.29 0.86 0.43 1.43 0.29 0.86 1.14 1.14 1.43 1.29 0.86 0.86 0.57 1.71	12.29 9.29 15.43 13.57

Table 7.2: Continuum path indicators for multiple target nodes $T = \{5, 6\}$ and different choices of α for network example in Fig. (7.1a)

7.5 Logical Flow

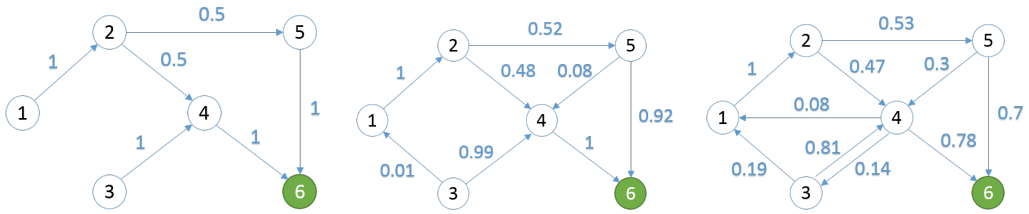
The transition probability matrix P used in routing algorithm (3) is derived from $P = D^{-1}A$. This transition probability yields $F_{sm}^{\{t,\bar{o}\}}$ which represents the aggregated stochastic flow of paths passing through node m (we call it stochastic flow of node m). In the stochastic flow of a path the degree of nodes located on paths matters: the paths composed of only low degree nodes dedicate higher stochastic flow to themselves than the ones including high degree nodes as well, since the transition probability of their edges are higher. This property has the advantage of picking paths with lower bottlenecks and reducing congestion in routing purposes.

However, if we set P to be a uniform distribution over the edges, namely $P_{ij} = \frac{A_{ij}}{d_{max}}$ where d_{max} is the maximum out-degree in the network, we can assign *logical* flows to

paths in which the probability of a path is only a function of its length not the degree of nodes located on that path. In this case, a node's flow represents the *fraction* of shortest paths passing through that node. For example, $F_{2,4}^{\{6,\bar{o}\}}(\alpha = 0.0001) = 0.5$ in table (7.3) implies that half of the shortest paths from node 2 to node 6 pass through node 4. Table (7.3) and Fig. (7.3) show the continuum for logical flows (setting a uniform P in the continuum algorithm). Note that for uniform P, since the probability of edges are restricted to $\frac{1}{d_{max}}$, even in the case of $\alpha = 1$ the path lengths $U^{\{6,\bar{o}\}}$ are not very large and still close to shortest paths distances (and far from hitting time distances).

α	$\mathrm{P}^{\{6\}}(\alpha)$	$F^{\{6,\bar{o}\}}(\alpha)$	$U^{\{6,\bar{o}\}}(\alpha)$
0.0001	0 1 0 0 0 0 0 0 0 0.5 0.5 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0	1 1 0 0.5 0.5 0 1 0 0.5 0.5 0 0 1 1 0 0 0 0 1 0 0 0 0 0 1	7 4 5 2 1
0.5	0 1 0 0 0 0 0 0 0 0.48 0.52 0 0.01 0 0 0.99 0 0 0 0 0 0 0 1 0 0 0 0.08 0 0.92 0 0 0 0 0 0	1 1 0 0.52 0.52 0 1 0 0.52 0.52 0.02 0.02 1 0.99 0.01 0 0 0 1 0 0 0 0 0.08 1	7.08 4.08 5.07 2.01 1.15
1	0 1 0 0 0 0 0 0 0 0.47 0.53 0 0.19 0 0 0.81 0 0 0.08 0 0.14 0 0 0.78 0 0 0 0.30 0 0.70 0 0 0 0 0 0	1.08 1.08 0.10 0.76 0.57 0.08 1.08 0.10 0.77 0.57 0.31 0.31 1.16 1.13 0.16 0.13 0.13 0.17 1.22 0.07 0.04 0.04 0.05 0.36 1.02	9.14 6.14 8.52 4.91 2.46

Table 7.3: Continuum path indicators for logical flow, target node $t = 6$, and different choices of α for network example in Fig. (7.1a)



(a) $\alpha = 0.0001$ (b) $\alpha = 0.5$ (c) $\alpha = 1$

Figure 7.3: Routing continuum for logical flow: routing edge probabilities $\mathrm{P}^{\{6\}}(\alpha)$ for network example in fig. (7.1a) with the same edge weights but uniform edge probabilities P, for different values of α which generates a continuum from shortest path to all path. Target is node 6.

7.6 A Novel Shortest Path Method

According to Theorem (11), once α goes to zero, the paths are pruned to *shortest* ones and $U_s^{\{t,\bar{o}\}}$ converges to shortest path distance from s to t. In the next chapter, we elaborate the behavior of proposed method for small α and how it can be exploited to devise a novel method for finding the shortest paths.

7.7 Network Measures Unification

Many network measures have been proposed in the literature for network analysis purposes [19], such as distance metrics for measuring the similarity (or diversity) between the nodes or entities of the network, centrality measures to assess a node's involvement or importance in the connectivity or communication between network entities, and topological indices to measure the compactness or resilience of the network against the failures. Commonly, these measures are founded based on either shortest path distances or hitting time distances. In this section, we show that the avoidance fundamental tensor in evaporation paradigm can yield a continuum of these metrics along with a more comprehensive view of them and provide a unifying framework for them.

We learned that the avoidance fundamental matrix in evaporating network $F^{\{t,\bar{o}\}}(\alpha)$ is a general form of avoidance fundamental matrix which simplifies to classical fundamental matrix when $\alpha = 1$. It represents the expected number of passages through nodes before reaching target t, for each choice of walk which is determined by α. Stacking up

avoidance fundamental matrices for all choices of target nodes forms the avoidance fundamental tensor $\boldsymbol{F}_{smt}(\alpha)$ $(= F_{sm}^{\{t,\bar{o}\}}(\alpha))$ with three dimensions: source node s, medial node m, and target node t (Fig. (7.4)). Aggregation of this tensor over its dimensions provides a unified framework for generating network measures. We particularly show that how the well-known measures can be written in terms of the avoidance fundamental tensor. Each of these network measures have their own implications and fit to different applications depending on the objective of the problem. Moreover, by other choices of α, we can generate different network defined on middle-length paths, i.e. paths that neither are confined to solely the shortest ones nor encompass all the random-walk paths. Fig. (7.4) visualizes the derivation of network measures from the avoidance fundamental tensor.

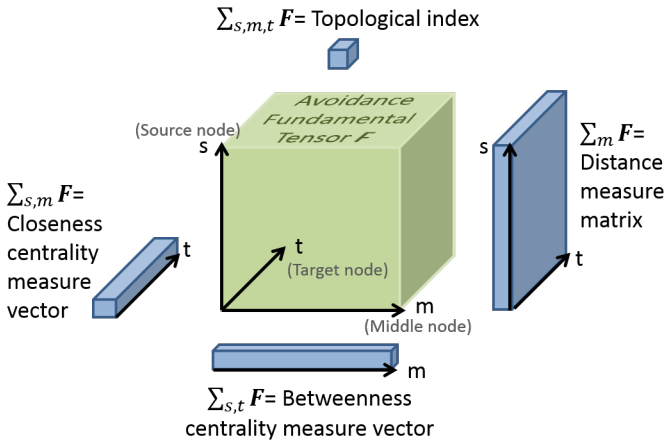


Figure 7.4: Network measures unification by avoidance fundamental tensor in a continuum framework

7.7.1 Distance Measure

As discussed earlier in this chapter, the avoidance hitting time in evaporating paradigm $H_s^{\{t,\bar{o}\}}(\alpha)$ yields a continuum from the shortest-path distance to all-path (hitting time) distance when α ranges from 0 to 1. Recall that $H_s^{\{t,\bar{o}\}}(\alpha)$ is in fact the aggregation of avoidance fundamental matrix in evaporating paradigm over the medial node m:

$H_s^{\{t,\bar{o}\}}(\alpha) = \sum_m F_{sm}^{\{t,\bar{o}\}}(\alpha)$. Hence, aggregating tensor $\boldsymbol{F}_{smt}(\alpha)$ over the medial nodes dimension returns a matrix of distance measures, ranging from shortest-path distance to hitting time distance with different choice of α, for any ordered pair of the nodes (Fig. (7.4)): $distance_{st}(\alpha) = \sum_m \boldsymbol{F}_{smt}(\alpha)$. In the case of weighted network, this aggregation would be weighted (Eq. (3.27)).

7.7.2 Closeness Measure

The centrality measures are categorized into two main types: Type I) based on volume measures, and Type II) based on distance measures [19]. Closeness measure is the closeness centrality of a node, which is from Type II (distance-based), is defined as the total distance of a node from the other nodes of the network. This implies how easily/closely the node is accessible/reachable from the other parts of the network and in brief how much in center of the network it is located.

Freeman's closeness centrality (or shortest-path closeness centrality) is the total shortest-path distance from a given node to all other nodes [52]. It can be represented as the aggregation of avoidance fundamental tensor in evaporating paradigm $\boldsymbol{F}_{smt}(\alpha)$ over the medial and target nodes dimensions, when $\alpha \to 0$: $closeness_s(0) = \sum_{m,t} \boldsymbol{F}_{smt}(\alpha \to 0)$. Since the closeness centrality technically has to measure closeness rather than farness, it is common to use the reciprocal form of the defined centrality measures, i.e. $\frac{1}{closeness_s(0)}$. On the other extreme, the random walk closeness centrality is proposed by Noh and Rieger [101] which is the inverse of the average hitting time distance to a given node from all other nodes. Ignoring the scalar factor of n, it is in fact the reciprocal form of aggregation of tensor $\boldsymbol{F}_{smt}(\alpha)$ over the source and medial nodes dimensions when $\alpha = 1$: $closeness_s(1) = \frac{n}{\sum_{s,m} \boldsymbol{F}_{smt}(\alpha=1)}$. Note that Noh and Rieger's closeness centrality measures the reachability of a given node from all other nodes which is an "authority" type of closeness. Equivalent "hub" type of closeness can be defined as the accessibility of a given node to all other nodes and be formulated as: $closeness_s(1) = \frac{n}{\sum_{m,t} \boldsymbol{F}_{smt}(\alpha=1)}$.

7.7.3 Betweenness Measure

Betweenness measure is a Type I (volume-based) centrality measure which quantifies the number of times a node acts as a bridge along the paths between different parts of

the network. The larger volume of these paths crossing that node, the more central the node is.

Recall that $F_{sm}^{\{t,\bar{o}\}}(\alpha \to 0)$ computed from a uniform transition probability matrix P indicates the fraction of shortest-paths from s to t that pass through m (Sec. 7.5). Therefore, aggregation of this metric over s gives the fraction of total shortest-paths from all the nodes to target t that m is located on. To find m's share of all shortest-path traffic between any pair of source and target, we aggregate avoidance fundamental tensor over the source and target nodes dimensions: $betweenness_m(0) = \sum_{s,t} \boldsymbol{F_{smt}}(\alpha \to 0)$. Metric $betweenness_m(0)$ is in fact the Freeman's *betweenness centrality* measure, called as shortest-path betweenness centrality too [53]. It measures the importance of node m in connectivity/bridging different parts of a network.

Different applications have inspired variations on betweenness centrality measures in literature. For instance, while inter-city trade might take only shortest paths for minimizing the costs, information usually flows across all possible paths. Thus modifying the betweenness centrality for random-walk (all-path), we obtain m's share of all possible walk between any pair of source and target: $betweenness_m(1) = \sum_{s,t} \boldsymbol{F_{smt}}(\alpha = 1)$. Newman [99] proposed a variation of random walk betweenness which is defined as the electrical (net) current flows through a medial node in an undirected network when a unit current flows through the network and aggregated over all pairs of vcc (source) and ground (target). We show that Newman's betweenness ($Nbetweenness$) can also be written as a function $f(.)$ of avoidance fundamental tensor aggregated over source and target nodes:

$$\begin{aligned} Nbetweenness_m &= \sum_{s,t} I(s \to m \to t) \\ &= \sum_{s,t} \sum_{k} \frac{1}{2} |\boldsymbol{F_{smt}}(\alpha = 1) P_{mk} - \boldsymbol{F_{skt}}(\alpha = 1) P_{km}| \\ &= \sum_{s,t} f(\boldsymbol{F_{smt}}(\alpha = 1)) \end{aligned} \tag{7.4}$$

We remark that if the network is directed and uni-directional, i.e. if $e_{ij} \in E$ then $e_{ji} \notin E$,

Newman's random walk betweenness centrality reduces to stationary probability:

$$\begin{aligned}\sum_{s,t} I(s \to m \to t) &= \sum_{s,t} \sum_{k \in \mathcal{N}(m)} |\boldsymbol{F_{smt}}(\alpha = 1) P_{mk}| \\ &= \sum_{s,t} \boldsymbol{F_{smt}}(\alpha = 1) \sum_{k \in \mathcal{N}(m)} P_{mk} \\ &= \sum_{s,t} \boldsymbol{F_{smt}}(\alpha = 1) = K\pi_m, \end{aligned} \tag{7.5}$$

where $\mathcal{N}(m)$ denotes the out-going neighbors of node m, K is the Kirchhoff index, and the last equation is based on Eq. (2.38).

Degree centrality can also be considered as a betweenness centrality in two ways: 1- enumerating only the 2-hop-length paths [19] and 2- enumerating total random walk paths between all pair-wise nodes. In the later case, the number of times that node m is crossed when going from node s to node t via a random walk path, aggregated over all s and t, is proportional to the degree of node m in an undirected network and proportional to the stationary probability of node m for the directed networks (Eq. (7.5)). Note that for the undirected networks the stationary probability is proportional to the degree: $\pi_m = \frac{d_m}{2|E|}$, where d_m is the degree of node m.

7.7.4 Topological Index

Topological indices are invariants calculated from graphs and express some information about the topology of the graphs. Topologocial indices are particularly important in mathematical chemistry since they can reflect some physical and chemical properties of the underlying molecular graph [117][116].

Wiener index is one of the well-known topological indices which has correlation to some important parameters of chemical species such as density, viscosity, surface tension, boiling point and other thermodynamic parameters [135]. Wiener index is defined as the summation of all pair-wise shortest path distances $distance_{st}(\alpha \to 0)$ which can be represented as the aggregation of avoidance fundamental tensor for $\alpha \to 0$ over all three dimensions:

$$W(G) = \sum_{s,t} distance_{st}(\alpha \to 0) = \sum_{s,m,t} \boldsymbol{F_{smt}}(\alpha \to 0) \tag{7.6}$$

Kirchhoff index is another well-known topological indices which is defined as the summation over the resistance distances of entire pairs of the nodes in the network [78]:

$K(G) = \frac{1}{2}\sum_{s,t}\Omega_{st}$. Kirchhoff index provides a measure of compactness (or robustness) of the network: the lower the value of $K(G)$ is the more compact the network G is. Kirchhoff index has an extensive application in molecular strength modeling in the mathematical chemistry literature [140, 103] as well as in linear algebra [14].

Tetali [126] showed that Effective resistance can be computed from commute time: $\Omega_{st} = \frac{1}{|\mathcal{E}|}C_{st}$, where undirected edges are considered as bidirectional edges and counted twice. Therefor, Kirchhoff index can be written in terms of commute time and be extended to directed networks as well: $K(G) = \frac{1}{2|\mathcal{E}|}\sum_{s,t}C_{st}$. The following relations show that how aggregation over all dimensions of fundamental tensor yields the Kirchhoff index:

$$K(G) = \frac{1}{2|\mathcal{E}|}\sum_{s,t}C_{st} = \frac{|\mathcal{V}|}{|\mathcal{E}|}\sum_{t}L^{+}_{tt} = \frac{1}{|\mathcal{E}|}\sum_{s,m,t}\boldsymbol{F_{smt}}(\alpha \to 1), \tag{7.7}$$

where the second equality comes from (2.42).

7.8 Proof of Theorems

Proof of Theorem 11. Let l_i's from countable set $\mathcal{C}$ be the length of walks from s to t such that $L_{st} = l_1 < l_2 < l_3 < ...$, and $\Pr_{l_i}$'s be the corresponding probabilities, where $\sum_{i=1}\Pr_{l_i} = 1$. The avoidance hitting cost (3.17) in evaporation network finds the following form:

$$U_s^{\{t,\bar{o}\}}(\alpha) = \frac{\sum_{i=1} l_i \alpha^{l_i}\Pr_{l_i}}{\sum_{i=1}\alpha^{l_i}\Pr_{l_i}}, \tag{7.8}$$

Proof of part (a)

When $\alpha \to 0$, the first term of numerator (and denominator) which is for $l_1 = L_{st}$ dominates the subsequent terms and $U_s^{\{t,\bar{o}\}}(\alpha)$ converges to $\frac{L_{st}\alpha^{L_{st}}\Pr_{L_{st}}}{\alpha^{L_{st}}\Pr_{L_{st}}} = L_{st}$.

Proof of part (b)

For $\alpha = 1$, there is no evaporation and network G_α splits into two disconnected subgraphs: the original network G with node t as its only absorbing node, and one isolated node which is node o. Then $U_s^{\{t,\bar{o}\}}(\alpha)$ reduces to the regular hitting cost from s to t in the original network G:

$$U_s^{\{t,\bar{o}\}}(\alpha = 1) = \frac{\sum_{i=1} l_i \Pr_{l_i}}{\sum_{i=1}\Pr_{l_i}} = \sum_{i=1} l_i \Pr_{l_i} = U_s^{\{t\}} \tag{7.9}$$

Proof of part (c)

We prove that if $\alpha_1 < \alpha_2$ then $U_s^{\{t,\bar{o}\}}(\alpha_1) \leq U_s^{\{t,\bar{o}\}}(\alpha_2)$, i.e.:

$$\frac{\sum_{i=1} l_i \alpha_1^{l_i} \Pr_{l_i}}{\sum_{i=1} \alpha_1^{l_i} \Pr_{l_i}} \leq \frac{\sum_{i=1} l_i \alpha_2^{l_i} \Pr_{l_i}}{\sum_{i=1} \alpha_2^{l_i} \Pr_{l_i}} \tag{7.10}$$

Cross-multiplying the fractions in (7.10), we compare the corresponding terms from the left-hand-side and right-hand-side polynomials. Without loss of generality assume that $i \leq j$:

$$\begin{aligned} (\alpha_2^{l_i} \Pr_{l_i})(l_j \alpha_1^{l_j} \Pr_{l_j}) + (\alpha_2^{l_j} \Pr_{l_j})(l_i \alpha_1^{l_i} \Pr_{l_i}) &\leq (\alpha_2^{l_i} \Pr_{l_i})(l_j \alpha_1^{l_j} \Pr_{l_j}) + (\alpha_2^{l_j} \Pr_{l_j})(l_i \alpha_1^{l_i} \Pr_{l_i}) \\ \Rightarrow \Pr_{l_i} \Pr_{l_j} (l_j \alpha_2^{l_i} \alpha_1^{l_j} + l_i \alpha_2^{l_j} \alpha_1^{l_i}) &\leq \Pr_{l_i} \Pr_{l_j} (l_j \alpha_1^{l_i} \alpha_2^{l_j} + l_i \alpha_1^{l_j} \alpha_2^{l_i}) \end{aligned}$$

Notice that for this inequality in two cases of: 1) $\Pr_{l_i}$ or $\Pr_{l_j}$ being zero, and 2) $i = j$ the equality holds; otherwise:

$$(l_j - l_i)\alpha_2^{l_i} \alpha_1^{l_j} < (l_j - l_i)\alpha_1^{l_i} \alpha_2^{l_j} \qquad \Longrightarrow \quad \alpha_1^{l_j - l_i} < \alpha_2^{l_j - l_i},$$

where the last inequality is obviously correct. □

Proof of Theorem 12. The avoidance fundamental matrix in evaporation network when the network is weighted finds the following form:

$$F_{sm}^{\{t,\bar{o}\}}(\alpha) = \frac{(\sum_{l_i = L_{sm}} \alpha^{l_i} \sum_{\zeta_j \in Z_{sm}(l_i)} \Pr_{\zeta_j}) \cdot (\sum_{l_i = L_{mt}} \alpha^{l_i} \sum_{\zeta_j \in Z_{mt}(l_i)} \Pr_{\zeta_j})}{\sum_{l_i = L_{st}} \alpha^{l_i} \sum_{\zeta_j \in Z_{st}(l_i)} \Pr_{\zeta_j}} \tag{7.11}$$

When $\alpha \to 0$, the first terms with lowest exponent of α dominate the subsequent terms and the equation above reduces to:

$$\lim_{\alpha \to 0} F_{sm}^{\{t,\bar{o}\}}(\alpha) = \lim_{\alpha \to 0} \frac{\alpha^{L_{sm} + L_{mt}} (\sum_{\zeta_j \in Z_{sm}(L_{sm})} \Pr_{\zeta_j}) . (\sum_{\zeta_j \in Z_{mt}(L_{mt})} \Pr_{\zeta_j})}{\alpha^{L_{st}} \sum_{\zeta_j \in Z_{st}(L_{st})} \Pr_{\zeta_j}} \tag{7.12}$$

Proof of part (a)

If m is not located on any shortest path from s to t, then $\alpha^{L_{sm}+L_{mt}} > \alpha^{L_{st}}$ and the limit in Eq. (7.12) converges to zero.

Proof of part (b)&(c)

If m is located on at least one of the shortest paths from s to t, then $\alpha^{L_{sm}+L_{mt}} = \alpha^{L_{st}}$ and the limit (7.12) has non-zero value: $\lim_{\alpha \to 0} F_{sm}^{\{t,\bar{o}\}}(\alpha) > 0$. On the other hand, we know that $\sum_{\zeta_j \in Z_{st}(L_{st})} \Pr_{\zeta_j} \geq (\sum_{\zeta_j \in Z_{sm}(L_{sm})} \Pr_{\zeta_j}) \cdot (\sum_{\zeta_j \in Z_{mt}(L_{mt})} \Pr_{\zeta_j})$ if

$L_{sm} + L_{mt} = L_{st}$. In the case that m is located on all of the shortest paths from s to t, it should be in L_{sm} distance from s and in L_{mt} distance to t on all of these paths (otherwise we can find a shorter path by connecting two shorter pieces) and thus we have: $\sum_{\zeta_j \in Z_{st}(L_{st})} \Pr_{\zeta_j} = (\sum_{\zeta_j \in Z_{sm}(L_{sm})} \Pr_{\zeta_j}) \cdot (\sum_{\zeta_j \in Z_{mt}(L_{mt})} \Pr_{\zeta_j})$ which results to $\lim_{\alpha \to 0} F_{sm}^{\{t,\bar{o}\}}(\alpha) = 1$. However, if m is not located on all of the shortest paths from s to t, then we have $\sum_{\zeta_j \in Z_{st}(L_{st})} \Pr_{\zeta_j} > (\sum_{\zeta_j \in Z_{sm}(L_{sm})} \Pr_{\zeta_j}) \cdot (\sum_{\zeta_j \in Z_{mt}(L_{mt})} \Pr_{\zeta_j})$ and so $\lim_{\alpha \to 0} F_{sm}^{\{t,\bar{o}\}}(\alpha) < 1$.

□

Proof of Theorem 7. We first prove that P is a transition probability matrix, namely is row stochastic:

$$\sum_{j \in \mathcal{N}(i)} \mathrm{P}_{ij} = \sum_{j \in \mathcal{N}(i)} P_{ij} \frac{Q_j^{\{T,\bar{o}\}}}{Q_i^{\{T,\bar{o}\}}} = \frac{1}{Q_i^{\{T,\bar{o}\}}} \sum_{j \in \mathcal{N}(i)} P_{ij} Q_j^{\{T,\bar{o}\}} = \frac{Q_i^{\{T,\bar{o}\}}}{Q_i^{\{T,\bar{o}\}}} = 1, \tag{7.13}$$

where the third equality is resulted because of Q is a harmonic function. Now we show that with the transformation in eq. (3.33) these equalities hold: $F_{sm}^{\{T,\bar{o}\}} = \mathrm{F}_{sm}^{\{T\}}$, $H_s^{\{T,\bar{o}\}} = \mathrm{H}_s^{\{T\}}$, and $U_s^{\{T,\bar{o}\}} = \mathrm{U}_s^{\{T\}}$.

$$\begin{aligned}
\mathrm{F}^{\{T\}} &= \sum_{k=0} \mathrm{P}_{\mathcal{T}\mathcal{T}}^k = \sum_{k=0} (Diag(Q^{\{T,\bar{o}\}})^{-1} P_{\mathcal{T}\mathcal{T}} Diag(Q^{\{T,\bar{o}\}}))^k \\
&= \sum_{k=0} Diag(Q^{\{T,\bar{o}\}})^{-1} P_{\mathcal{T}\mathcal{T}}^k Diag(Q^{\{T,\bar{o}\}}) \\
&= Diag(Q^{\{T,\bar{o}\}})^{-1} (\sum_{k=0} P_{\mathcal{T}\mathcal{T}}^k) Diag(Q^{\{T,\bar{o}\}}) \\
&= Diag(Q^{\{T,\bar{o}\}})^{-1} F^{\{T,o\}} Diag(Q^{\{T,\bar{o}\}}) \\
&= F^{\{T,\bar{o}\}}
\end{aligned}$$

For the hitting times we have $\mathrm{H}_s^{\{T\}} = \mathrm{F}^{\{T\}} \mathbf{1}$ and $H_s^{\{T,\bar{o}\}} = F^{\{T,\bar{o}\}} \mathbf{1}$, so $H_s^{\{T,\bar{o}\}} = \mathrm{H}_s^{\{T\}}$.

The following relations also hold for hitting costs:

$$\begin{aligned} U_s^{\{T,\bar{o}\}} &= \sum_m F_{sm}^{\{T,\bar{o}\}} r_m^{\{T,\bar{o}\}} \\ &= \sum_m F_{sm}^{\{T,\bar{o}\}} \sum_j \frac{Q_j^{\{T,\bar{o}\}}}{Q_m^{\{T,\bar{o}\}}} P_{mj} w_{mj} \\ &= \sum_m F_{sm}^{\{T,\bar{o}\}} \sum_j P_{mj} w_{mj} \\ &= \sum_m F_{sm}^{\{T,\bar{o}\}} r_m \\ &= \sum_m F_{sm}^{\{T\}} r_m \\ &= U_s^{\{T\}}, \end{aligned}$$

where the first and third equalities are based on (3.27) and (3.33) respectively. □

Chapter 8

SSSP Queries and Distance Oracles

8.1 Introduction

Shortest path algorithms are of great importance in many fields and applications. Single-source shortest path (SSSP) and all-pair shortest path (APSP) form two main types of shortest path problems in which the shortest path from one source node to all the other nodes and between all the pairs of nodes are computed respectively. However, particular applications might require something in the middle: answering several SSSP queries but not APSP. In such cases, an algorithm with prepossessing time faster than APSP and query time faster than SSSP sounds more attractive than the existing SSSP and APSP algorithms. Computing the shortest distance from national airports to reach to main international hubs and finding the shortest sequence of friends to connect to celebrities in social networks are examples of need for faster and less complex algorithms than APSP.

We have developed an algorithm that answers SSSP queries in general directed and weighted networks (without negative cycles) in $O(m+n)$ time with $O(n^2)$ space requirement and $O(n^\omega)$ preprocessing time, where n is the number of nodes, m is the number of edges, and $\omega < 2.376$ is the current exponent for the fast matrix multiplication. The best time complexity for directed and weighted networks belongs to Bellman-Ford algorithm with $O(mn)$ for SSSP [49, 13] and to Pettie's [107] with $O(mn + n^2 loglogn)$ for APSP. The query time of our algorithm is enormously faster than SSSP algorithms and the prepossessing computations requires lower time complexity than APSP algorithms for dense network with $m > n^{1.376}$. We elaborate our method in Section (8.3).

8.2 Related work

Single-source shortest path (SSSP) and all-pair shortest path (APSP) are two well-known shortest path problems which have been studied vastly over 60 years. In the most general types of network, i.e. directed and real weighted networks, the best time-complexity algorithm belongs to Bellman-Ford algorithm with $O(mn)$ for SSSP [49, 13] and to Pettie's [107] with $O(mn + n^2 loglogn)$ for APSP. However, the idea of pre-computing a *distance oracle* which is faster than APSP to answer to shortest path queries with time-complexity lower than SSSP, was first proposed by Thorup and Zwick [128]. Their distance oracle supports the weighted and undirected networks and returns an approximated distance with a stretch of $2k-1$, a preprocess time of $O(mn^{1/k})$, space requirement of $O(n^{1+1/k})$, and query time of $O(1)$ for every fixed integer $k \geq 1$. In a follow-up work, Yuster and Zwick [138], proposed an algorithm for answering distance queries in directed and integer-weighted network with absolute value at most M using fast matrix multiplication. By preprocessing time of $\tilde{O}(Mn^{\omega})$, where ω is the exponent of fast matrix multiplication, and space requirement of $O(n^2)$, their algorithm answers to a single distance query in $O(n)$ time. In our algorithm, we improve the query time to $O(m)$ for an SSSP query which consists of $n-1$ distance queries.

8.3 Method Overview

According to Theorem (11), once α goes to zero, the paths are pruned to *shortest* ones and $U_s^{\{t,\bar{o}\}}$ converges to shortest path distance L_{st}. However, we didn't mention how to find the shortest path and how small α should be in practice. In this chapter, we propose an algorithm to answer shortest path distance queries efficiently and a method to determine α accordingly.

First, the error $\epsilon_{st}(\alpha) = U_s^{\{t,\bar{o}\}}(\alpha) - L_{st}$ is formulated in terms of α in Section (8.4) to provide a better understanding of convergence behavior of avoidance hitting cost to shortest path distance when α decreases. Afterwards in Section (8.5), we prove that if the errors become all smaller than δ/d, our algorithm in (14) finds the shortest paths from all nodes to a single target t (SSSP). Here, δ is the largest value by which all the edge weights are divisible and d is the out-degree of nodes. We derive a theoretical bound for α to make errors smaller than δ/d and demonstrate that the bound is tight for special designed

networks (Sections (8.6.1),(8.6.2)). However, for real-world networks and generative models the required α is not that restricted and can be picked considerably larger than the theoretical bound (Section (8.6.3)). Therefore, Section (8.7) is dedicated to devising a high performance machine learning method to recommend the required α for real networks. This method uses network *local* features, such as node degrees, clustering coefficient, and network assortativity, to train a boosted decision tree and yields an α as a *global* feature for the network. For this training we have used 55 real networks and generative models.

8.4 Convergence Behavior and the Corresponding Error

In this section, we formulate error $\epsilon_{st}(\alpha) = U_s^{\{t,\bar{o}\}}(\alpha) - L_{st}$ in terms of α to study the convergence behavior of avoidance hitting cost to shortest path distance when α goes to 0. This formulation enables us later in this chapter to find a bound for α and make the error be smaller than δ/d.

Let l_i's from countable set $\mathcal{C}$ be the length of walks from s to t such that $L_{st} = l_1 < l_2 < l_3 < ...$, and $\Pr_{l_i}$'s be the corresponding probabilities (if there are more than one walk with the same length, the Pr is the aggregated probability of the walks). Since δ is divisible by all walk lengths l_i, we can assume that any two consecutive walk lengths differ by δ, i.e. $l_{i+1} = l_i + \delta$, otherwise we can always add a walk length with zero-probability, i.e. $\Pr_{l_i} = 0$. For unweighted networks $\delta = 1$. In the evaporating network, every edge e_{ij} is assigned a multiplicative factor of $\alpha^{w_{ij}}$ and so walks with length of l_i have the total probability of $\alpha^{l_i}\Pr_{l_i}$. Recall that $0 \leq \Pr_{l_i} \leq 1$ and $\sum_{l_i} \Pr_{l_i} = 1$. Then, the avoidance hitting cost can be decomposed into the shortest path distance plus an error term:

$$
\begin{aligned}
U_s^{\{t,\bar{o}\}}(\alpha) &= \frac{\sum_{i=1} l_i \alpha^{l_i} \mathrm{Pr}_{l_i}}{\sum_{i=1} \alpha^{l_i} \mathrm{Pr}_{l_i}} \\
&= \frac{L_{st} \sum_{i=1} \alpha^{l_i} \mathrm{Pr}_{l_i} + \sum_{i=2} (l_i - l_{i-1}) \sum_{k=i} \alpha^{l_k} \mathrm{Pr}_{l_k}}{\sum_{i=1} \alpha^{l_i} \mathrm{Pr}_{l_i}} \\
&= L_{st} + \delta \frac{\sum_{i=2} \sum_{k=i} \alpha^{l_k} \mathrm{Pr}_{l_k}}{\sum_{i=1} \alpha^{l_i} \mathrm{Pr}_{l_i}} \\
&= L_{st} + \delta \sum_{j=1} \alpha^{j\delta} \frac{\sum_{i=1} \alpha^{l_i} \mathrm{Pr}_{l_{i+j}}}{\sum_{i=1} \alpha^{l_i} \mathrm{Pr}_{l_i}} \\
&= L_{st} + \delta \sum_{j=1} \alpha^{j\delta} \gamma_j \\
&= L_{st} + \epsilon_{st}(\alpha), \qquad (8.1)
\end{aligned}
$$

It can be seen that $\epsilon_{st}(\alpha)$ is a non-negative function of α and so always $U_s^{\{t,\bar{o}\}}(\alpha) \geq L_{st}$, meaning that avoidance hitting cost converges to shortest path distance from above. In the next part, we show that by putting $\epsilon_{st}(\alpha) < \delta/d_s$ and computing the inverse function, a bound for α can be found.

8.5 Finding the Edges on the Shortest Path

Beside finding the shortest path distance by computing $U_s^{\{t,\bar{o}\}}(\alpha)$ for small enough α, we need to find the path itself. In the following theorem, we show how to find the successor of each node in the shortest path tree and specify the edges located on the shortest path.

Theorem 14 (Shortest Path Routing Strategy)**.** *Let $\epsilon_{st}(\alpha) < \delta/d_s$, where d_s is the number of out-going neighbors of s and δ is the largest value by which all the edge weights are divisible. s's out-going edge with highest probability, i.e. $P_{sj} = \max_m P_{sm}$, is located on the shortest path from s to t.*

Corollary 10. *In an unweighted network, if $\epsilon_{st}(\alpha) < 1/d_s$, then node j with maximum Q among the neighbors, i.e. $Q_j^{\{t,\bar{o}\}} = \max_{m \in \mathcal{N}_{out}(s)} Q_m^{\{t,\bar{o}\}}$, is located on the shortest path from s to t.*

Since, finding the shortest path is a recursive process, the whole path can be obtained by finding the successor of each node via the highest edge probability $P_{sj} = \max_m P_{sm}$

in each step, starting from s and until getting to t. The following algorithm summarizes the shortest path routing strategy based on the proposed method

Algorithm 5 ALL PAIR SHORTEST PATH

input: Probability transition matrix P, weight matrix W, and α

output: Shortest paths

$P(\alpha) = \alpha^W \odot P$

$F^{\{o\}}(\alpha) = (I - P(\alpha))^{-1}$

for each target t **do**

$\forall e_{ij} \in E : \mathrm{P}_{ij}^{\{t\}}(\alpha) = P_{ij}(\alpha)\frac{F_{jt}^{\{o\}}(\alpha)}{F_{it}^{\{o\}}(\alpha)}$

$\forall i \in V : \mathrm{successor}\{i\} = \arg\max_j \mathrm{P}_{ij}^{\{t\}}(\alpha)$

shortest-path tree rooted at $t = \cup_{i \in V} e_{i,\mathrm{successor}\{i\}}$

end for

8.6 Bound for α

8.6.1 Theoretical bound

We find a bound for α to make distance error ϵ smaller than δ/d_{max}. For this purpose, we first find an upper bound for γ to obtain an upper bound for distance error. Recall that $\epsilon = \delta \sum_{j=1} \alpha^{j\delta}\gamma_j$:

$$\begin{aligned}\gamma_j &= \frac{\sum_{i=1}\alpha^{l_i}\mathrm{Pr}_{l_{i+j}}}{\sum_{i=1}\alpha^{l_i}\mathrm{Pr}_{l_i}} \leq \frac{\alpha^{l_1}\sum_{i=1}\mathrm{Pr}_{l_{i+j}}}{\sum_{i=1}\alpha^{l_i}\mathrm{Pr}_{l_i}} \leq \frac{\alpha^{l_1}(1-(\sum_{i=1}^{j}\mathrm{Pr}_{l_i}))}{\alpha^{l_1}\mathrm{Pr}_{l_1}} \\ &\leq \frac{\alpha^{l_1}(1-\mathrm{Pr}_{l_1})}{\alpha^{l_1}\mathrm{Pr}_{l_1}} = \frac{1-\mathrm{Pr}_{l_1}}{\mathrm{Pr}_{l_1}} \leq \frac{1-(\frac{1}{d_{max}})^{L_{max}}}{(\frac{1}{d_{max}})^{L_{max}}}, \end{aligned} \tag{8.2}$$

where $L_{max} = \max_{(s,t)} L_{st}$ is the diameter of the network and $d_{max} = \max_i d_i$ is the maximum out-degree in the network. The last inequality is resulted from the worst case scenario in which the shortest path probability Pr_{l_1} is composed of multiplication of least edge probabilities, i.e. $\frac{1}{d_{max}}$, and for the longest distance of network diameter. The upper bound for distance error ϵ_{st} is obtained as follows:

$$\epsilon_{st} \leq \delta \sum_{i=1} \alpha^{i\delta}\left(\frac{1-(\frac{1}{d_{max}})^{L_{max}}}{(\frac{1}{d_{max}})^{L_{max}}}\right) = \delta\frac{\alpha^{\delta}}{1-\alpha^{\delta}}\left(\frac{1-(\frac{1}{d_{max}})^{L_{max}}}{(\frac{1}{d_{max}})^{L_{max}}}\right) \tag{8.3}$$

To guarantee that the distance error is smaller than δ/d_{max}, we can make its upper bound (8.3) be lower than δ/d_{max}, i.e. $\delta\frac{\alpha^\delta}{1-\alpha^\delta}(\frac{1-(\frac{1}{d_{max}})^{L_{max}}}{(\frac{1}{d_{max}})^{L_{max}}}) < \frac{\delta}{d_{max}}$. Now, we can find a bound for α in terms of δ, network diameter L_{max}, and maximum out-degree d_{max} to have distance error ϵ smaller than δ/d_{max}:

$$\alpha \leq (\frac{1}{(d_{max})^{L_{max}+1} - d_{max} + 1})^{1/\delta} \approx (\frac{1}{d_{max}})^{(L_{max}+1)/\delta} \tag{8.4}$$

A similar bound can be achieved with a completely different approach for the special case of unweighted networks which is presented in (8.8).

8.6.2 Tightness of the bound

We show that the bound in (8.4) is not a loose bound and can be actually achieved for a general network. According to Theorem (14), edge with max P being located on shortest path is the necessary condition for having $\epsilon < \delta/d$. To confirm the tightness of α's bound (8.4), we design a network to challenge this necessary condition. The idea is to build a chain of nodes with maximum possible Q and one chain with minimum possible Q. Now if a node has two neighbors, one from maximum-Q-chain but further from target and one from minimum-Q-chain but closer to target, the neighbor closer to target node should have higher Q.

Small example

In figure (8.1) the upper row of blue nodes form the maximum-Q-chain whose Q values are labeled above the nodes. The lower row form the minimum-Q-chain whose values are answers of the following linear system of equations. Recall that Q is a harmonic function and the Q value of each node is the (weighted) average of its neighbors' values. In these figures, the target node is represented by green color, the source node by orange

color, and the evaporation node is omitted to avoid cluttering.

$$\begin{aligned}
\text{(Tier 1): } q_1 &= \frac{\alpha}{d} + \alpha\frac{d-1}{d}q_4 \\
\text{(Tier 2): } q_2 &= \frac{\alpha}{d}q_1 + \alpha\frac{d-1}{d}q_4 \\
\text{(Tier 3): } q_3 &= \frac{\alpha}{d}q_2 + \alpha\frac{d-1}{d}q_4 \\
\text{(Tier 4): } q_4 &= \frac{\alpha}{d}q_3 + \alpha\frac{d-1}{d}q_4,
\end{aligned} \tag{8.5}$$

where d is 3 in this example and tiers represent the distance to target (diameter L is equal to 4 here).

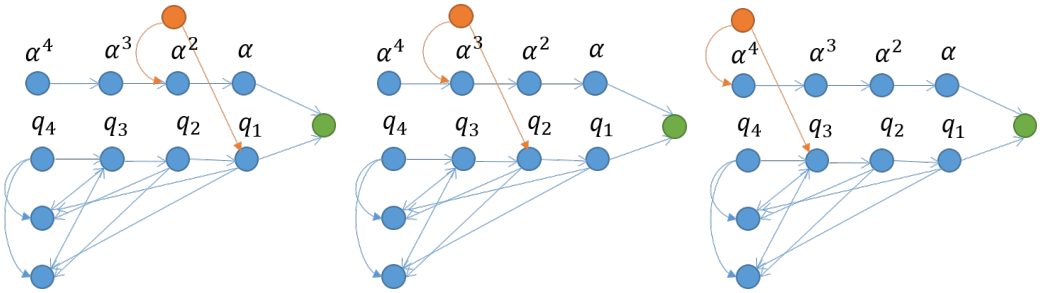
In order that the necessary condition for $\epsilon < \delta$ be satisfied (10) for all the three cases in figure (8.1), the following inequalities should hold:

$$\begin{aligned}
q_1 &> \alpha^2 \\
q_2 &> \alpha^3 \\
q_3 &> \alpha^4
\end{aligned} \tag{8.6}$$

Solving linear system (8.5) and substituting the answers in inequalities (8.6) the following bounds for α are obtained:

$$\begin{aligned}
\frac{\alpha}{3} > \alpha^2 &\rightarrow \alpha \lesssim \frac{1}{3} \\
(\frac{\alpha}{3})^2 + \frac{2(\frac{\alpha}{3})^5 - 2(\frac{\alpha}{3})^6}{1-\alpha+2(\frac{\alpha}{3})^5} > \alpha^3 &\rightarrow \alpha \lesssim (\frac{1}{3})^2 \\
(\frac{\alpha}{3})^3 + \frac{2(\frac{\alpha}{3})^5 - 2(\frac{\alpha}{3})^7}{1-\alpha+2(\frac{\alpha}{3})^5} > \alpha^4 &\rightarrow \alpha \lesssim (\frac{1}{3})^3,
\end{aligned} \tag{8.7}$$

where the last inequality is the most restrictive bound for α.



(a) case 1 (b) case 2 (c) case 3

Figure 8.1: Three designed networks with $d_{max} = 3$ and $L = 4$ to challenge the necessary condition in (10) and confirm the tightness of α's bound (8.4).

General form

We can generalize this design for a network with d as its maximum out-degree, L as the diameter (number of tiers), and any number of nodes and edges. The linear system of minimum Q's (8.5) finds the following look:

$$\mathbf{q} = M\mathbf{q} + \mathbf{b} \rightarrow \mathbf{q} = (I - M)^{-1}\mathbf{b}, \tag{8.8}$$

where $\mathbf{q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ \vdots \\ q_{L-1} \\ q_L \end{bmatrix}$, $M = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & \alpha\frac{d-1}{d} \\ \frac{\alpha}{d} & 0 & 0 & \dots & 0 & \alpha\frac{d-1}{d} \\ 0 & \frac{\alpha}{d} & 0 & \dots & 0 & \alpha\frac{d-1}{d} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & \alpha\frac{d-1}{d} \\ 0 & 0 & 0 & \dots & \frac{\alpha}{d} & \alpha\frac{d-1}{d} \end{bmatrix}$, and $\mathbf{b} = \begin{bmatrix} \frac{\alpha}{d} \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$. Therefor, the necessary condition (10) is satisfied if:

$$\begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ \vdots \\ q_{L-2} \\ q_{L-1} \end{bmatrix} > \begin{bmatrix} \alpha^2 \\ \alpha^3 \\ \alpha^4 \\ \vdots \\ \alpha^{L-1} \\ \alpha^L \end{bmatrix}, \tag{8.9}$$

where the inequality is entry-wise.

Preposition 1. *The solution of (8.8) is* $q_j = (\frac{\alpha}{d})^j + \frac{(d-1)(\frac{\alpha}{d})^{L+1} - (d-1)(\frac{\alpha}{d})^{L+j}}{1-\alpha+(d-1)(\frac{\alpha}{d})^{L+1}}$ *which if substituted in (8.9) results in the bound of* $\alpha \lesssim (\frac{1}{d})^{L-1}$ *for* α.

Proof. To compute $(I-M)^{-1}$, we split matrix M into two matrices:

$$M = M_1 + M_2 = \begin{bmatrix} 0 & 0 & 0 & ... & 0 & 0 \\ \frac{\alpha}{d} & 0 & 0 & ... & 0 & 0 \\ 0 & \frac{\alpha}{d} & 0 & ... & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & ... & 0 & 0 \\ 0 & 0 & 0 & ... & \frac{\alpha}{d} & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & ... & 0 & \alpha\frac{d-1}{d} \\ 0 & 0 & 0 & ... & 0 & \alpha\frac{d-1}{d} \\ 0 & 0 & 0 & ... & 0 & \alpha\frac{d-1}{d} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & ... & 0 & \alpha\frac{d-1}{d} \\ 0 & 0 & 0 & ... & 0 & \alpha\frac{d-1}{d} \end{bmatrix}, \quad (8.10)$$

where M_1 is a Nilpotent matrix with index L (i.e. $M_1^k = 0$ for $k \geq L$) and M_2 is a rank 1 matrix. Due to the Nilpotent property [96] of M_1 we can easily write $(I-M_1)^{-1}$ as the expansion of powers of M_1 up to L, and the rank 1 property of M_2 enables us to benefit from Sherman-Morrison formula [119]:

$$(I-M)^{-1} = ((I-M_1)-M_2)^{-1} = (I-M_1)^{-1} + \frac{(I-M_1)^{-1}uv'(I-M_1)^{-1}}{1-v'(I-M_1)^{-1}u}. \quad (8.11)$$

The components of (8.11) are computed as follows:

$$(I-M_1)^{-1} = I + M_1 + M_1^2 + ... + M_1^{L-1} \quad (8.12)$$

$$= \begin{bmatrix} 1 & 0 & 0 & ... & 0 & 0 \\ \frac{\alpha}{d} & 1 & 0 & ... & 0 & 0 \\ (\frac{\alpha}{d})^2 & \frac{\alpha}{d} & 1 & ... & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ (\frac{\alpha}{d})^{L-2} & (\frac{\alpha}{d})^{L-3} & (\frac{\alpha}{d})^{L-4} & ... & 1 & 0 \\ (\frac{\alpha}{d})^{L-1} & (\frac{\alpha}{d})^{L-2} & (\frac{\alpha}{d})^{L-3} & ... & \frac{\alpha}{d} & 1 \end{bmatrix} \quad (8.13)$$

$$M_2 = uv' = \begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & ... & 0 & \alpha\frac{d-1}{d} \end{bmatrix} \quad (8.14)$$

$$
\begin{aligned}
(I-M_1)^{-1}u &= \begin{bmatrix} 1 & 0 & 0 & ... & 0 & 0 \\ \frac{\alpha}{d} & 1 & 0 & ... & 0 & 0 \\ (\frac{\alpha}{d})^2 & \frac{\alpha}{d} & 1 & ... & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ (\frac{\alpha}{d})^{L-2} & (\frac{\alpha}{d})^{L-3} & (\frac{\alpha}{d})^{L-4} & ... & 1 & 0 \\ (\frac{\alpha}{d})^{L-1} & (\frac{\alpha}{d})^{L-2} & (\frac{\alpha}{d})^{L-3} & ... & \frac{\alpha}{d} & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} 1 \\ 1+\frac{\alpha}{d} \\ 1+\frac{\alpha}{d}+(\frac{\alpha}{d})^2 \\ \vdots \\ \sum_{i=0}^{L-2}(\frac{\alpha}{d})^i \\ \sum_{i=0}^{L-1}(\frac{\alpha}{d})^i \end{bmatrix} \qquad (8.15)
\end{aligned}
$$

$$
\begin{aligned}
v'(I-M_1)^{-1} &= \begin{bmatrix} 0 & 0 & 0 & ... & 0 & \alpha\frac{d-1}{d} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & ... & 0 & 0 \\ \frac{\alpha}{d} & 1 & 0 & ... & 0 & 0 \\ (\frac{\alpha}{d})^2 & \frac{\alpha}{d} & 1 & ... & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ (\frac{\alpha}{d})^{L-2} & (\frac{\alpha}{d})^{L-3} & (\frac{\alpha}{d})^{L-4} & ... & 1 & 0 \\ (\frac{\alpha}{d})^{L-1} & (\frac{\alpha}{d})^{L-2} & (\frac{\alpha}{d})^{L-3} & ... & \frac{\alpha}{d} & 1 \end{bmatrix} \\
&= (d-1).\begin{bmatrix} (\frac{\alpha}{d})^L & (\frac{\alpha}{d})^{L-1} & (\frac{\alpha}{d})^{L-2} & ... & (\frac{\alpha}{d})^2 & \frac{\alpha}{d} \end{bmatrix} \qquad (8.16)
\end{aligned}
$$

$$
\begin{aligned}
& v'(I-M_1)^{-1}u \\
= & \left[\begin{array}{cccccc} 0 & 0 & 0 & ... & 0 & \alpha\frac{d-1}{d} \end{array}\right]
\left[\begin{array}{cccccc} 1 & 0 & 0 & ... & 0 & 0 \\ \frac{\alpha}{d} & 1 & 0 & ... & 0 & 0 \\ (\frac{\alpha}{d})^2 & \frac{\alpha}{d} & 1 & ... & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ (\frac{\alpha}{d})^{L-2} & (\frac{\alpha}{d})^{L-3} & (\frac{\alpha}{d})^{L-4} & ... & 1 & 0 \\ (\frac{\alpha}{d})^{L-1} & (\frac{\alpha}{d})^{L-2} & (\frac{\alpha}{d})^{L-3} & ... & \frac{\alpha}{d} & 1 \end{array}\right]
\left[\begin{array}{c} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \\ 1 \end{array}\right] \\
= & (d-1)\sum_{i=1}^{L}(\frac{\alpha}{d})^i \qquad (8.17)
\end{aligned}
$$

$$
(I-M)^{-1}b = (\frac{\alpha}{d})(\left[\begin{array}{c} 1 \\ \frac{\alpha}{d} \\ (\frac{\alpha}{d})^2 \\ \vdots \\ (\frac{\alpha}{d})^{L-2} \\ (\frac{\alpha}{d})^{L-1} \end{array}\right] + \frac{(d-1)(\frac{\alpha}{d})^L}{1-(d-1)\sum_{i=1}^{L}(\frac{\alpha}{d})^i}\left[\begin{array}{c} 1 \\ 1+\frac{\alpha}{d} \\ 1+\frac{\alpha}{d}+(\frac{\alpha}{d})^2 \\ \vdots \\ \sum_{i=0}^{L-2}(\frac{\alpha}{d})^i \\ \sum_{i=0}^{L-1}(\frac{\alpha}{d})^i \end{array}\right]) \qquad (8.18)
$$

$$
\begin{aligned}
\Longrightarrow \mathbf{q} &= \left[\begin{array}{c} \frac{\alpha}{d} \\ (\frac{\alpha}{d})^2 \\ (\frac{\alpha}{d})^3 \\ \vdots \\ (\frac{\alpha}{d})^{L-1} \\ (\frac{\alpha}{d})^L \end{array}\right] + \frac{(d-1)(\frac{\alpha}{d})^{L+1}}{1-\alpha+(d-1)(\frac{\alpha}{d})^{L+1}}\left[\begin{array}{c} 1-\frac{\alpha}{d} \\ 1-(\frac{\alpha}{d})^2 \\ 1-(\frac{\alpha}{d})^3 \\ \vdots \\ 1-(\frac{\alpha}{d})^{L-1} \\ 1-(\frac{\alpha}{d})^L \end{array}\right] \qquad (8.19) \\
&\approx \left[\begin{array}{c} \frac{\alpha}{d} \\ (\frac{\alpha}{d})^2 \\ (\frac{\alpha}{d})^3 \\ \vdots \\ (\frac{\alpha}{d})^{L-1} \\ (\frac{\alpha}{d})^L \end{array}\right], \qquad (8.20)
\end{aligned}
$$

Substituting $\mathbf{q}$ in (8.9):

$$\begin{bmatrix} \frac{\alpha}{d} \\ (\frac{\alpha}{d})^2 \\ (\frac{\alpha}{d})^3 \\ \vdots \\ (\frac{\alpha}{d})^{L-2} \\ (\frac{\alpha}{d})^{L-1} \end{bmatrix} > \begin{bmatrix} \alpha^2 \\ \alpha^3 \\ \alpha^4 \\ \vdots \\ \alpha^{L-1} \\ \alpha^L \end{bmatrix}, \tag{8.21}$$

where the most restrictive bound for α is resulted from the last inequality, i.e. $\alpha < (\frac{1}{d})^{L-1}$. Note that all the material in this part can be easily extended to the weighted case as well. □

8.6.3 α in *real* networks and generative models

In spite of the tightness of α's bound (8.4) for the designed networks in (8.6.2), we show that α does not require to be that small to achieve $\epsilon < \delta/d$ in real networks and is way larger in practice. Apposite to the designed network (8.1), where maximum out degree d and diameter L can be independently large, in real networks and generative models these two network metrics are not independent from each other and topologically it is almost impossible that both d and L be very large. In addition, the topology of the network and affinity of the nodes are in a way that the situations in (8.1) are very rare to happen; thus, the necessary condition (14) is not much challenged and the required alpha is far from the bound (8.4). Figure (8.2) illustrates this phenomena that the required α in real networks and generative models is much larger than the bound (blue line).

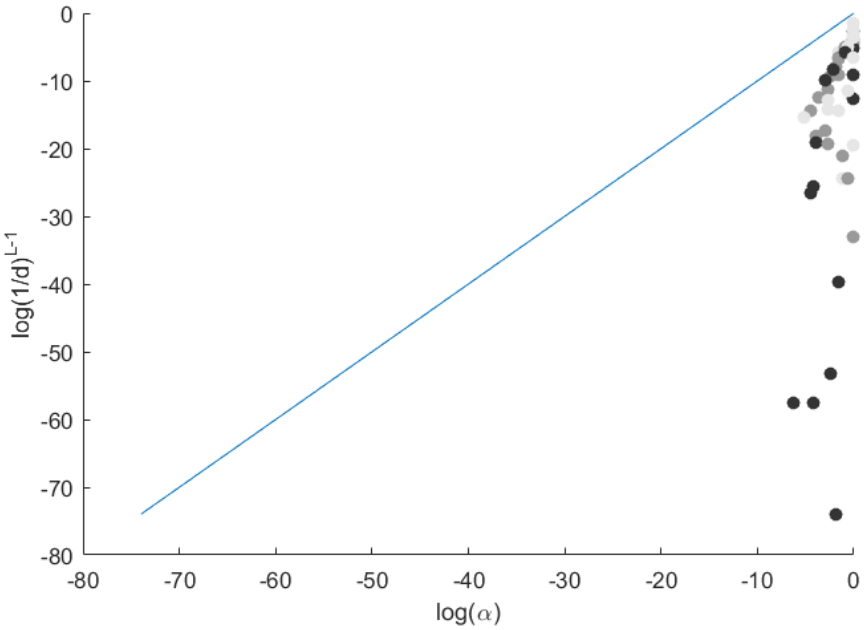


Figure 8.2: The required α for $\epsilon < \delta/d$ in 55 real networks and generative models. Darker color implies higher size of the network.

For this experiment, we computed the largest possible α for 55 networks consisting of real networks, such as Facebook [90], western states power grid of the United States [132], coauthorship network of scientists [97], and political blogs network [3], and generative models, such as small world model [132], preferential attachment [12], Erdos Reney random model [45], and Kronecker random and core-periphery [86]. Largest "possible" α means an α that satisfies $\epsilon < \delta/d$ for all pair of shortest paths. We find α iteratively by starting from an initial value and increment it if the inequality $\epsilon < \delta/d$ holds and decrement it if otherwise, until the inequality changes.

8.7 α recommender module

We develop a recommender module to recommend an α for any inputed network based on the network local features or metrics. For this purpose, we leverage machine learning methods to learn α from a set of network metrics which are computationally less complex

than $O(n^{\omega})$. The following list is the set of network metrics which have been used as the features (a mixture of binary-valued, integer-valued, and continuous-valued attributes) of the machine learning method:

1. Weighted network (1) or unweighted network (0)
2. Directed network (1) or undirected network (0)
3. Strongly connected (1) or not (0)
4. Number of nodes n
5. Number of edges m
6. Maximum out-degree
7. Minimum out-degree
8. Mean of out-degrees $\bar{d} = E(d)$
9. Variance of out-degrees $\sigma^2(d)$
10. Maximum of neighbors' out-degree differences dd_{max}, where degree difference for each node i's neighbors is defined as $dd_i = \max_{j \in \mathcal{N}_i} d_j - \min_{j \in \mathcal{N}_i} d_j$
11. Minimum of neighbors' out-degree differences dd_{min}
12. Mean of neighbors' out-degree differences $\overline{dd}$
13. Variance of neighbors' out-degree differences $\sigma^2(dd)$
14. Maximum clustering coefficient C_{max}, where local clustering coefficient for each node i is defined as $C_i = \frac{|\{e_{jk}: v_j, v_k \in \mathcal{N}_i, e_{jk} \in \mathcal{E}\}|}{d_i(d_i-1)}$ [132]
15. Minimum clustering coefficient C_{min}
16. Mean of clustering coefficients $\bar{C}$
17. Variance of clustering coefficients $\sigma^2(C)$
18. Maximum core number cn_{max} [9]

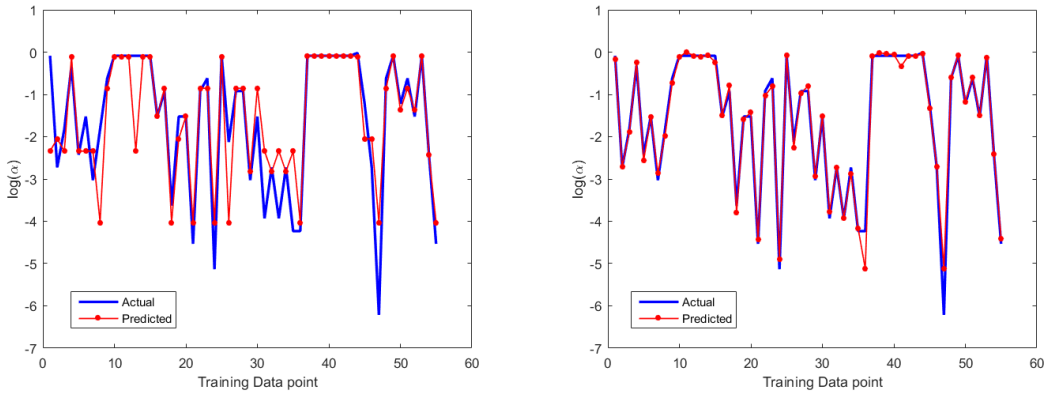
19. Mean of core numbers $\bar{cn}$

20. Network assortativity r [98].

We applied boosted decision tree with gradient boosting strategy on data and received a prominent performance, which is reported in terms of root mean square error (RMSE), mean absolute error (MAE), and mean one-sided error (MOSE) in table (8.1). In MOSE only if predicted α is higher than actual α is counted as an error since as long as $\alpha_{predicted} \leq \alpha_{actual}$ the computed shortest path in our method is error-less. The boosted decision tree is trained with 500 simple trees. Figure (8.3) illustrates the superiority of the trained boosted decision tree performance compared against the decision tree trained on the same data.

Table 8.1: Boosted decision tree performance trained on 55 networks data.

	RMSE	MAE	MOSE
Training error	0.2126	0.1071	0.0535
Cross validation error	1.0329	0.7952	0.4048



(a) Decision Tree (b) Boosted Decision Tree

Figure 8.3: The performance comparison of boosted decision tree with gradient boosting strategy against decision tree trained on 55 network data.

8.8 Proof of Theorems

Proof of Theorem 14. We first find an expression for distance error $\epsilon_{st}(\alpha)$ in terms of edge costs and probabilities. According to Theorem (7), any avoidance paradigm can be transformed to a corresponding classical paradigm, and we have: $P_{ij}(\alpha) = P_{ij}(\alpha)\frac{Q_j^{\{t,\bar{o}\}}(\alpha)}{Q_i^{\{t,\bar{o}\}}(\alpha)}$, $F_{sm}^{\{t\}}(\alpha) = F_{sm}^{\{t,\bar{o}\}}(\alpha)$, and $U_s^{\{t\}}(\alpha) = U_s^{\{t,\bar{o}\}}(\alpha)$. In the transformed classical paradigm, we can write the hitting cost in the recursive form and transform it back to corresponding avoidance metrics. (Just note that for the rest of the proof, we drop α's to avoid clutter and make the relations more readable):

$$
\begin{aligned}
U_s^{\{t\}} &= r_s + \sum_{m\in\mathcal{N}_{out}(s)} P_{sm} U_m^{\{t\}} \\
\rightarrow U_s^{\{t,\bar{o}\}} &= r_s + \sum_{m\in\mathcal{N}_{out}(s)} P_{sm} U_m^{\{t,\bar{o}\}} \\
&= \sum_{m\in\mathcal{N}_{out}(s)} P_{sm} w_{sm} + \sum_{m\in\mathcal{N}_{out}(s)} P_{sm} U_m^{\{t,\bar{o}\}} \\
&= P_{sj}(w_{sj} + U_j^{\{t,\bar{o}\}}) + \sum_{m\neq j} P_{sm}(U_m^{\{t,\bar{o}\}} + w_{sm}) \\
\rightarrow L_{st} + \epsilon_{st} &= P_{sj}(w_{sj} + L_{jt} + \epsilon_{jt}) + \sum_{m\neq j} P_{sm}(L_{mt} + \epsilon_{mt} + w_{sm})
\end{aligned}
$$

Out-going edge set of node s can be divided into two subset of $\mathcal{J}_e$ and $\mathcal{J}_e^C$, where $\mathcal{J}_e$ consists of the edges that are located on the shortest path from s to t, and $\mathcal{J}_e^C$ is the complementary set. Let $\mathcal{J}_v$ be the corresponding out-going neighbors to $\mathcal{J}_e$, i.e. $\mathcal{J}_e = \cup_{i\in\mathcal{J}_v} e_{si}$ and $|\mathcal{J}_e| = |\mathcal{J}_v|$. We prove that the edge with highest probability $P_{sj} = \max_m P_{sm}$ belong to $\mathcal{J}_e$. If $\mathcal{J}_e$ includes all of s's out-going edges and $\mathcal{J}_e^C = \emptyset$, the proof is complete; otherwise, there exists at least one s's out-going edge which is not located on the shortest path from s to t, i.e. $|\mathcal{J}_e| \leq d_s - 1$. Now, we show that the maximum edge probability in set $\mathcal{J}_e$ is higher than the maximum edge probability in

$\mathcal{J}_e^C$:

$$\begin{aligned}
\rightarrow \epsilon_{st} &= (\sum_{j\in\mathcal{J}_v} \text{P}_{sj} - 1)L_{st} + \sum_{j\in\mathcal{J}_v} \text{P}_{sj}\epsilon_{jt} + \sum_{m\notin\mathcal{J}_v} \text{P}_{sm}(L_{mt} + \epsilon_{mt} + w_{sm}) \\
&\geq (\sum_{j\in\mathcal{J}_v} \text{P}_{sj} - 1)L_{st} + \sum_{m\notin\mathcal{J}_v} \text{P}_{sm}(L_{mt} + w_{sm}) \\
&\geq (\sum_{j\in\mathcal{J}_v} \text{P}_{sj} - 1)L_{st} + \sum_{m\notin\mathcal{J}_v} \text{P}_{sm}(L_{st} + \delta) \\
&= (\sum_{j\in\mathcal{J}_v} \text{P}_{sj} - 1)L_{st} + (1 - \sum_{j\in\mathcal{J}_v} \text{P}_{sj})(L_{st} + \delta) \\
&= (1 - \sum_{j\in\mathcal{J}_v} \text{P}_{sj})\delta \qquad (8.22)
\end{aligned}$$

Substituting the lower bound of ϵ_{st} (8.22) in the Theorem's assumption of $\epsilon_{st} < \delta/d_s$, the following result is obtained:

$$(1 - \sum_{j\in\mathcal{J}_v} \text{P}_{sj})\delta < \delta/d_s \rightarrow \sum_{j\in\mathcal{J}_v} \text{P}_{sj} > \frac{d_s - 1}{d_s} \rightarrow \sum_{j\in\mathcal{J}_v^C} \text{P}_{sj} < \frac{1}{d_s}, \qquad (8.23)$$

which proves that the highest edge probability in $\mathcal{J}_e$ is at least equal to $\frac{1}{d_s}$, while the highest edge probability in $\mathcal{J}_e^C$ is strictly less than $\frac{1}{d_s}$.

□

Proof of Corollary 10. In unweighted networks $\delta = 1$ and P_{sj} for all $j \in \mathcal{N}_{out}(s)$ are equal. Therefor, according to Theorem (14) edge e_{sj} with maximum P is located on the shortest path from s to t:

$$P_{sj}\frac{Q_j^{\{t,\bar{o}\}}}{Q_s^{\{t,\bar{o}\}}} = \max_m P_{sm}\frac{Q_m^{\{t,\bar{o}\}}}{Q_s^{\{t,\bar{o}\}}} \qquad (8.24)$$

$$\rightarrow \quad j = \arg\max_m Q_m^{\{t,\bar{o}\}} \qquad (8.25)$$

□

Error bound in terms of α for unweighted networks.

$$\tilde{A} = D^{-1/2}AD^{-1/2} = D^{1/2}PD^{-1/2} \rightarrow \tilde{A}_{\mathcal{TT}} = D^{1/2}P_{\mathcal{TT}}D^{-1/2} \qquad (8.26)$$

$$\tilde{A}_{\mathcal{TT}} = U\Lambda U^T \rightarrow P_{\mathcal{TT}} = D^{-1/2}U\Lambda U^T D^{1/2} \qquad (8.27)$$

$$P^l_{\mathcal{TT}} = D^{-1/2} U \Lambda^l U^T D^{1/2} \rightarrow (P^l_{\mathcal{TT}})_{ij} = \sum_k \lambda_k^l \sqrt{\frac{d_j}{d_i}} U_{ki} U_{kj} \tag{8.28}$$

$$H_s^{\{t\}}(\alpha) = \frac{\sum_{l=l_0} l [P^{l-1}_{\mathcal{TT}}(\alpha) P_{\mathcal{TA}}(\alpha)]_{st}}{\sum_{l=l_0} [P^{l-1}_{\mathcal{TT}}(\alpha) P_{\mathcal{TA}}(\alpha)]_{st}} \tag{8.29}$$

$$\begin{aligned}
\sum_{l=l_0} l [P^{l-1}_{\mathcal{TT}}(\alpha) P_{\mathcal{TA}}(\alpha)]_{st} &= \sum_{l=l_0} l\alpha^l [P^{l-1}_{\mathcal{TT}} P_{\mathcal{TA}}]_{st} && (8.30)\\
&= \sum_k (\sum_{l=l_0} l\alpha^l \lambda_k^{l-1}) [(D^{-1/2} U_k U_k^T D^{1/2}) P_{\mathcal{TA}}]_{st} && (8.31)\\
&= \sum_k (\sum_{l=l_0} l\alpha^l \lambda_k^{l-1}) Z_{st}(k) && (8.32)\\
&= \sum_k \frac{\partial}{\partial \lambda_k} (\sum_{l=l_0} (\alpha\lambda_k)^l) Z_{st}(k) && (8.33)\\
&= \sum_k \frac{\partial}{\partial \lambda_k} (\frac{(\alpha\lambda_k)^{l_0}}{1-\alpha\lambda_k}) Z_{st}(k) && (8.34)\\
&= \sum_k (\frac{l_0 (\alpha\lambda_k)^{l_0-1}\alpha}{1-\alpha\lambda_k} + \frac{(\alpha\lambda_k)^{l_0}\alpha}{(1-\alpha\lambda_k)^2}) Z_{st}(k) && (8.35)\\
&= \sum_k (\frac{l_0 (\alpha\lambda_k)^{l_0-1}\alpha}{1-\alpha\lambda_k}) Z_{st}(k) + \sum_k (\frac{(\alpha\lambda_k)^{l_0}\alpha}{(1-\alpha\lambda_k)^2}) Z_{st}(k) && (8.36)\\
&= l_0 \sum_k (\sum_{l=l_0} \alpha^l \lambda_k^{l-1}) Z_{st}(k) + \sum_k (\frac{(\alpha\lambda_k)^{l_0}\alpha}{(1-\alpha\lambda_k)^2}) Z_{st}(k) && (8.37)\\
&= l_0 \sum_{l=l_0} \sum_k \alpha^l \lambda_k^{l-1} [(D^{-1/2} U_k U_k^T D^{1/2}) P_{\mathcal{TA}}]_{st} + \sum_k (\frac{(\alpha\lambda_k)^{l_0}\alpha}{(1-\alpha\lambda_k)^2}) Z_{st}(k) \\
&= l_0 \sum_{l=l_0} [P^{l-1}_{\mathcal{TT}}(\alpha) P_{\mathcal{TA}}(\alpha)]_{st} + \sum_k (\frac{(\alpha\lambda_k)^{l_0}\alpha}{(1-\alpha\lambda_k)^2}) Z_{st}(k) && (8.38)
\end{aligned}$$

$$\rightarrow H_{st}(\alpha) = l_0 + \frac{\sum_k (\frac{(\alpha\lambda_k)^{l_0}\alpha}{(1-\alpha\lambda_k)^2}) Z_{st}(k)}{\sum_{l=l_0} [P^{l-1}_{\mathcal{TT}}(\alpha) P_{\mathcal{TA}}(\alpha)]_{st}} \tag{8.39}$$

To bound error $H_s^{\{t\}}(\alpha) - l_0$, we have the following relations:

$$
\begin{aligned}
H_s^{\{t\}}(\alpha) - l_0 &= \frac{\sum_k (\frac{(\alpha\lambda_k)^{l_0}\alpha}{(1-\alpha\lambda_k)^2}) Z_{st}(k)}{\sum_{l=l_0} [P_{\mathcal{TT}}^{l-1}(\alpha) P_{\mathcal{TA}}(\alpha)]_{st}} && (8.40)\\
&= \frac{\sum_k (\frac{(\alpha\lambda_k)^{l_0}\alpha}{(1-\alpha\lambda_k)^2}) [(D^{-1/2} U_k U_k^T D^{1/2}) P_{\mathcal{TA}}]_{st}}{\sum_{l=l_0} [P_{\mathcal{TT}}^{l-1}(\alpha) P_{\mathcal{TA}}(\alpha)]_{st}} && (8.41)\\
&= \frac{\sum_{j\in\mathcal{N}(t)} \sum_k (\frac{(\alpha\lambda_k)^{l_0}\alpha}{(1-\alpha\lambda_k)^2}) U_{ks} U_{kj} \sqrt{\frac{d_j}{d_s}} \frac{a_{jt}}{d_j}}{\sum_{j\in\mathcal{N}(t)} \sum_{l=l_0} [P_{\mathcal{TT}}^{l-1}(\alpha)]_{sj} [P_{\mathcal{TA}}(\alpha)]_{jt}} && (8.42)\\
&= \frac{\sum_{j\in\mathcal{N}(t)} \sum_k (\frac{(\alpha\lambda_k)^{l_0}\alpha}{(1-\alpha\lambda_k)^2}) U_{ks} U_{kj} \sqrt{\frac{d_j}{d_s}} \frac{a_{jt}}{d_j}}{\sum_{j\in\mathcal{N}(t)} F_{sj}(\alpha) \frac{\alpha a_{jt}}{d_j}} && (8.43)\\
&\leq \max_{j\in\mathcal{N}(t)} \frac{\sum_k (\frac{(\alpha\lambda_k)^{l_0}\alpha}{(1-\alpha\lambda_k)^2}) U_{ks} U_{kj} \sqrt{\frac{d_j}{d_s}} \frac{a_{jt}}{d_j}}{F_{sj}(\alpha) \frac{\alpha a_{jt}}{d_j}} && (8.44)\\
&= \max_{j\in\mathcal{N}(t)} \frac{\sum_k (\frac{(\alpha\lambda_k)^{l_0}}{(1-\alpha\lambda_k)^2}) U_{ks} U_{kj} \sqrt{\frac{d_j}{d_s}}}{F_{sj}(\alpha)} && (8.45)\\
&\leq \max_{j\in\mathcal{N}(t)} \frac{\sqrt{\frac{d_j}{d_s}} \sum_k \frac{|\alpha\lambda_k|^{l_0}}{(1-\alpha\lambda_k)^2} |U_{ks} U_{kj}|}{F_{sj}(\alpha)} && (8.46)\\
&\leq \max_{j\in\mathcal{N}(t)} \frac{\sqrt{\frac{d_j}{d_s}} \frac{(\alpha\lambda_1)^{l_0}}{1-\alpha\lambda_1} \sum_k \frac{1}{1-\alpha\lambda_k} |U_{ks} U_{kj}|}{F_{sj}(\alpha)} && (8.47)\\
&\leq \max_{j\in\mathcal{N}(t)} \frac{\sqrt{\frac{d_j}{d_s}} \frac{(\alpha\lambda_1)^{l_0}}{1-\alpha\lambda_1} (\sum_k \frac{1}{1-\alpha\lambda_k} U_{ks}^2)^{\frac{1}{2}} (\sum_k \frac{1}{1-\alpha\lambda_k} U_{kj}^2)^{\frac{1}{2}}}{F_{sj}(\alpha)} && (8.48)\\
&= \max_{j\in\mathcal{N}(t)} \frac{\sqrt{\frac{d_j}{d_s}} \frac{(\alpha\lambda_1)^{l_0}}{1-\alpha\lambda_1} (F_{ss}(\alpha))^{\frac{1}{2}} (F_{jj}(\alpha))^{\frac{1}{2}}}{F_{sj}(\alpha)} && (8.49)\\
&= \max_{j\in\mathcal{N}(t)} \frac{\sqrt{\frac{d_j}{d_s}} \frac{(\alpha\lambda_1)^{l_0}}{1-\alpha\lambda_1} (F_{ss}(\alpha))^{\frac{1}{2}} (F_{jj}(\alpha))^{\frac{1}{2}}}{(\frac{d_j}{d_s} F_{js}(\alpha))^{\frac{1}{2}} (F_{sj}(\alpha))^{\frac{1}{2}}} && (8.50)\\
&= \max_{j\in\mathcal{N}(t)} \frac{\frac{(\alpha\lambda_1)^{l_0}}{1-\alpha\lambda_1}}{(G_{js}(\alpha))^{\frac{1}{2}} (G_{sj}(\alpha))^{\frac{1}{2}}} && (8.51)\\
&\leq \frac{\frac{(\alpha\lambda_1)^{l_0}}{1-\alpha\lambda_1}}{(\alpha P_{min})^{l_0-1}} && (8.52)\\
&= \frac{\alpha\lambda_1}{1-\alpha\lambda_1} (\frac{\lambda_1}{P_{min}})^{l_0-1} && (8.53)\\
&< \frac{\alpha}{1-\alpha} (\frac{1}{P_{min}})^{l_0-1} && (8.54)\\
&= \frac{\alpha}{1-\alpha} (d_{max})^{l_0-1} && (8.55)
\end{aligned}
$$

Equation (25) is resulted from the following relation in undirected networks:

$$F_{js} = F_{sj}\frac{d_s}{d_j} \tag{8.56}$$

□

Chapter 9

Replacement Path and Distance Sensitivity Oracles

9.1 Introduction

Shortest path problem has long been one of the fundamental problems in computer science and is exploited in many fields such as shortest path routing in wireless networks, protein interaction analysis, transportation problems handling, social networks studies, and VLSI design. Although numerous fast and scalable algorithms have been developed for *static* shortest path problem over the past decades, devising low-cost *dynamic* shortest path algorithms to efficiently find the shortest paths after the changes in the network is still considered to be challenging.

Due to this demand for developing more flexible algorithms to support the changes in the network, several problems have been posed under different names and objectives. In the *replacement paths* problem, the objective is to answer to query (s, t, f) by computing the shortest replaced path efficiently from a fixed source node s to a fixed target node t for avoiding each of the nodes (or edges) located on the shortest path denoted by f. The more general forms of this problem are for multiple sources by answering queries $(*, t, f)$ and the all pairs replacement paths format which answers queries $(*, *, f)$ by efficiently finding the shortest replaced path for all pairs of source and target nodes, while avoiding an arbitrary failed node (or edge) f and by constructing a *distance sensitivity oracle*. To be even more advanced is to find the replacement path in case of *multiple*

failures and answer to corresponding query $(s, t, \mathcal{F})$, which is very challenging and is still considered as an open problem. The main applications for distance sensitivity oracle are routing in failure-prone networks, Vickrey price problem, and finding k shortest simple paths. When a network is prone to failures, it is very expensive to compute the shortest paths every time from the scratch. Distance sensitivity oracle provides this privilege to compute the new shortest paths faster and with lower cost. In extension, fault-tolerant routing protocol is a distributed solution which seeks for the shortest route avoiding the set of failures while trying to optimize the amount of memory stored in the routing tables of the nodes (compact routing scheme) [127]. In the Vickrey price problem from auction theory [67] the edges of a networks are each owned by a selfish agent and the objective is to determine the value of an edge according to how difficult it gets to route the information in the network if that edge fails. This can be done by benefiting from sensitivity distance oracle to compare the shortest path length before and after deleting the edge [16]. This problem is closely related to find the most damaging or vital node (or edge) in the network [36]. Moreover, k shortest simple paths can be easily computed by running k executions of a replacement paths algorithm [44].

In this chapter, we propose a novel and simple-to-implement replacement path algorithm to support *multiple* failures with arbitrary size and answer to $(*, t, \mathcal{F})$ queries efficiently as long as the size of failure $|\mathcal{F}|$ is constant and not growing with the size of network n. This algorithm is founded upon two developed concepts: avoidance Markov chain and evaporation paradigm. The advantage of our algorithm is multiple folds:

1. By leveraging from fast matrix multiplication (with exponent ω, which is currently $\omega = 2.376$ [134]), the sensitivity distance oracle with size $O(n^2)$ is constructed in $O(n^\omega)$ time. This oracle answers to distance and path queries $(*, t, \mathcal{F})$ in only $O(m)$ time, where n is the number of nodes and m is the number of edges.

2. In contrast to the existing work, the proposed sensitivity distance oracle does not depend on failure size and can be exploited for any size of the failure once is constructed.

3. The algorithm supports the general directed networks with arbitrary weights (without negative cycle).

4. The algorithm can be simply modified to support edge failures as well as find alternative longer paths.

Therefor, the method presented in this section gives an affirmative answer to two questions of Bernstein and Karger [16] in the conclusion part of their paper: "*We cannot really hope to improve upon the static version, but can we make the oracle dynamic: if we delete a single vertex, can we do better than constructing another oracle from scratch? Also, can we efficiently handle more than one vertex failure at a time?*".

9.2 Related work

Sensitivity distance oracle algorithms have been studied vastly for supporting the single failure case. For weighted and directed networks, Demetrescu et al. [39] proposed an $O(n^2 \log n)$-size oracle which is constructed in $O(mn^2 + n^3 \log n)$ time and answers to shortest path length queries (s, t, f) in $O(1)$ time. Bernstein and Karger [16] improved the previous algorithm by lower construction time of $O(mn \log^2 n + n^2 \log^3 n)$ but space size of $O(n^2 \log^2 n)$ and the same query time of $O(1)$. The same authors also presented a randomized algorithm [15] which is improved in construction time and storage size with a factor of $\log n$ compared to their deterministic algorithm and the same query time. Note that the query time for *finding* the shortest path is $O(L)$ in all of these algorithms where L is the length of the path. The approximate algorithm proposed by Khanna and Baswana [76] provides a lower storage requirement of $O(kn^{1+1/k}\frac{log^3 n}{\epsilon^4})$ for unweighted and undirected networks. This algorithm returns $(2k-1)(1+\epsilon)$-approximate distance query in $O(k)$ time for given an integer $k > 1$ and a fraction $\epsilon > 0$.

As one of the first attempts to support more than one failure, Duan and Pettie [42] proposed a method for covering the dual-failures ($f = 2$). Their method requires the storage size of $O(n^2 \log^3 n)$ which is constructed in polynomial time. The query time for returning the length of shortest path is $O(\log n)$ and for returning the whole path is $O(L \log n)$. According to the authors, this method cannot be extended to cases with $f > 2$, since it becomes very complex and requires $O(n^f \log^3 n)$ of space. The other f-sensitivity distance oracle is a $(8k-2)(f+1)$-approximate algorithm suggested by Chechik et al. [28] to support more than two failures $f > 2$ for undirected networks. The oracle is constructed in polynomial time and takes

$O(fkn^{1+1/k}\log(nW))$ of space to answer distance queries. The query time for this algorithm is $O(|\mathcal{F}|.\log^2 n.\log\log n.\log\log L)$, where $\mathcal{F}$ is the number of failures and $\mathcal{F} < f$, W is the weight of heaviest edge, and L is the longest distance in the network. Weimann and Yuster [134] propose a randomized algorithm for constructing a sensitivity distance oracle with size of $\tilde{O}(n^{3-\alpha})$ given a trading-off parameter $0 < \alpha < 1$ and conditioned on the failure order being $|\mathcal{F}| = O(\frac{\log n}{\log\log n})$. Notation $\tilde{O}$ indicates that some $\log n$ has been dropped from the order. This algorithm was originally devised for integer-weighted graphs with edge weights chosen from $\{-W, ..., +W\}$ [133] and then was extended to real-weighted graphs in a follow-up work [134]. For the case of integer weights, the construction time is $O(Wn^{1+\omega-\alpha})$ with query time of $\tilde{O}(n^{2-(1-\alpha)/|\mathcal{F}|})$, and the real weights case has been become possible by construction time of $O(n^{4-\alpha})$ and query time of $\tilde{O}(n^{2-2(1-\alpha)/|\mathcal{F}|})$. The authors take advantage of the fast matrix multiplication, with ω as the exponent, in their computations which is currently $\omega = 2.376$ [134]. Note that both of the reviewed works for supporting multiple failures require to know the size of failure $|\mathcal{F}|$ in advance for their oracle construction.

9.3 Method Overview

The replacement path method is constructed based on the theory developed in past two chapters. In Chapter (7), we demonstrated that once α goes to zero in an evaporating network, the paths are pruned to *shortest* ones and $U_s^{\{t,\bar{o}\}}$ converges to shortest path distance from s to t in the original network (Theorem 11). Then we showed in Chapter (8) that how to find α to make error $\epsilon_{st}(\alpha) = U_s^{\{t,\bar{o}\}}(\alpha) - L_{st}$ less than δ/d and presented the corresponding shortest path algorithm (5). The only gap here is that we had those theories for cases with no failures while here we have to exclude a set of failed nodes in our shortest path algorithm. In this chapter, we show that this gap is filled with a theorem that we prove (15): $U_s^{\{t,\overline{\mathcal{F}},\bar{o}\}}(\alpha)$ converges to shortest path distance from s to t *excluding failure part of the network, i.e. set* $\mathcal{F}$, once α goes to zero. Henceforth, a similar technique can be followed to find the replacement path: for each failure set $\mathcal{F}$, the fundamental matrix $F^{\{\mathcal{F},o\}}$ is computed from $F^{\{o\}}$ efficiently (Theorem 1) and the edge probabilities are computed for a single target. Edges with highest probability resides on the shortest path. We illustrate this replacement path algorithm in Alg. (6)

and discuss the corresponding complexity time in Section (9.4.1).

9.4 Theoretical Framework and Complexity Discussion

Theorem 15. *Assume set $\mathcal{F}$ of nodes have failed in weighted network G. If $\alpha \to 0$ in the corresponding evaporating network G_α, the avoidance hitting cost $U_s^{\{t,\overline{\mathcal{F}},\bar{o}\}}(\alpha)$ in G_α converges to shortest-path distance in G where failure set $\mathcal{F}$ is discarded from the network.*

$$\lim_{\alpha \to 0} U_s^{\{t,\overline{\mathcal{F}},\bar{o}\}}(\alpha) = L_{st}^{\{\overline{\mathcal{F}}\}} \tag{9.1}$$

Algorithm (6) finds the replacement shortest paths efficiently from all nodes to target t while there are a set of failure nodes $\mathcal{F}$ in the network.

Algorithm 6 Replacement path algorithm for all sources to single target queries with multiple failures $(*, t, \mathcal{F})$

Input:

Probability transition matrix P, weight matrix W, and α

Output:

Shortest paths from all nodes to single target t which do not pass any nodes in failure set $\mathcal{F}$

Preprocess:

$P(\alpha) = \alpha^W \odot P$

$F^{\{o\}}(\alpha) = (I - P(\alpha))^{-1}$

Query: $(*, t, \mathcal{F})$

Query response:

$M = (F^{\{o\}}_{\mathcal{F},\mathcal{F}}(\alpha))^{-1}$

$\forall i \in V : F^{\{\mathcal{F},o\}}_{i,t}(\alpha) = F^{\{o\}}_{i,t}(\alpha) - F^{\{o\}}_{i,\mathcal{F}}(\alpha) M F^{\{o\}}_{\mathcal{F},t}(\alpha)$

$\forall e_{ij} \in E : \mathrm{P}^{\{t\}}_{ij}(\alpha) = P_{ij}(\alpha) \frac{F^{\{\mathcal{F},o\}}_{jt}(\alpha)}{F^{\{\mathcal{F},o\}}_{it}(\alpha)}$

$\forall i \in V : \text{successor}\{i\} = \arg\max_j \mathrm{P}^{\{t\}}_{ij}(\alpha)$

Shortest-path tree rooted at $t = \cup_{i \in V} e_{i,\text{successor}\{i\}}$

where the second equation in Query response is resulted from Theorem (1) and the third one is a substitution of (2) in Theorem (14).

9.4.1 Preprocess time and space

The purpose of preprocess part is to compute and store $F\{^o\}(\alpha)$ which can be used to answer replacement path queries very efficiently. The required space for storing this matrix is n^2 where n is the number of nodes. Regarding the complexity time, the inverse computation is the main costly component with complexity time of $O(n^\omega)$, where $\omega = 2.376$, and is discussed in the following. Recall that the ReccomenderModule requires 20 network metrics as input who are all less complex than $O(n^\omega)$.

Matrix Inverse: The computational complexity of matrix multiplication of two $n \times n$ matrices is sub-qubic; according to Strassen algorithm [122] the complexity is $O(n^{2.807})$ and later on it reduced even more to $O(n^{2.376})$ by Coppersmith-Winograd algorithm [35]. Cormen et al. [37] proved that inversion is no harder than multiplication (Theorem 28.2). A divide and conquer algorithm that uses blockwise inversion to invert a matrix runs with the same time complexity as the matrix multiplication algorithm that is used internally.

9.4.2 Query time

For having a fast query time, we leverage from the incremental computation in Theorem (1). Based on this theorem, only an $O(1)$-computation is required to compute $F_{i,t}^{\{\mathcal{F},o\}}(\alpha)$ from precomputed matrix $F^{\{o\}}(\alpha)$ and for any given failure set $\mathcal{F}$, as long as $|\mathcal{F}|$ is constant with respect to network size n. In this case, computation of term $M = (F_{\mathcal{F},\mathcal{F}}^{\{o\}}(\alpha))^{-1}$ requires $O(|\mathcal{F}|^\omega)$ time which is still considered $O(1)$. The most costly component of query computations is computing the new probabilities $\mathrm{P}_{ij}^{\{t\}}(\alpha)$ for all edges which takes $O(m)$ time.

9.5 Proof of Theorems

Proof of Theorem 15. Let G be an unweighted network and avoidance hitting time $H_s^{t,\{\overline{\mathcal{F},o}\}}(\alpha)$ is defined on the corresponding evaporation paradigm G_α, and $\mathcal{F}$ is the set of failure nodes. We write the avoidance hitting time in terms of transition probabilities

(3.6):

$$H_s^{\{t,\overline{\mathcal{F}},\bar{o}\}}(\alpha) = \frac{\sum_{k=k_1} k[[P(\alpha)]_{\mathcal{T}_2\mathcal{T}_2}^{k-1}[P(\alpha)]_{\mathcal{T}_2\mathcal{A}_2}]_{st}}{\sum_{k=k_1}[[P(\alpha)]_{\mathcal{T}_2\mathcal{T}_2}^{k-1}[P(\alpha)]_{\mathcal{T}_2\mathcal{A}_2}]_{st}} = \frac{\sum_{k=k_1} k\alpha^k[P_{\mathcal{T}_1\mathcal{T}_1}^{k-1}P_{\mathcal{T}_1\mathcal{A}_1}]_{st}}{\sum_{k=k_1}\alpha^k[P_{\mathcal{T}_1\mathcal{T}_1}^{k-1}P_{\mathcal{T}_1\mathcal{A}_1}]_{st}}, \tag{9.2}$$

where $P(\alpha)$ is the transition matrix of evaporation network and P belongs to the original network. In the original network G the target node t as well as the failure set $\mathcal{F}$ form the absorbing set: $\mathcal{A}_1 = \{t\} \cup \mathcal{F}$ and $\mathcal{T}_1 = V \setminus \mathcal{A}_1$. In the evaporating network G_α, the evaporation node o is absorbing too: $\mathcal{A}_2 = \{o\} \cup \mathcal{A}_1$ and $\mathcal{T}_2 = V \setminus \mathcal{A}_2$. When $\alpha \to 0$

$$\lim_{\alpha\to 0} H_s^{\{t,\overline{\mathcal{F}},\bar{o}\}}(\alpha) = \lim_{\alpha\to 0} \frac{\sum_{k=k_1} k\alpha^k[P_{\mathcal{T}_1\mathcal{T}_1}^{k-1}P_{\mathcal{T}_1\mathcal{A}_1}]_{st}}{\sum_{k=k_1}\alpha^k[P_{\mathcal{T}_1\mathcal{T}_1}^{k-1}P_{\mathcal{T}_1\mathcal{A}_1}]_{st}} = k_1, \tag{9.3}$$

k_1 is the smallest number of steps to take from s to reach t in the transient part of G, which interprets the shortest path distance from s to t excluding the nodes in $\mathcal{F}$.

For the weighted network, the proof is straightforward following the same idea for the unweighted network as well as using Theorem (11). □

Chapter 10

Conclusion

In this dissertation, we presented our research on complex network analysis under three subjects of cascade, reachability, and routing. For these studies, we developed a platform of powerful theories and tools founded on Markov chain theory and random walk methods which supports the general *weighted* and *directed* networks.

In Chapter 2, we reviewed certain Markov chain classical metrics and showed how to compute them in a unified way. We also collected and proved a library of useful lemmas and theorems for these metrics and their relations to each other which were used in applications such as finding the most influential people in a social network for influence maximization, devising an oracle to efficiently answer dynamic reachability queries, and computing the articulation points of directed networks in later chapters.

In Chapter 3, we developed and introduced Markov chain avoidance metrics which provide more flexibility in the design of Markov chain and impose new conditions on the transition to avoid (or transit) a specific state (or a set of states) before the stopping criteria. We established the usefulness of these theories through applications such as proposing a pivotality metric to rank the importance of nodes in reachabilities, developing a generative model for a routing continuum from shortest path to (random walk) all path, and devising a distance oracle which answers to single-source shortest path (SSSP) queries, and finds replacement paths in multiple failures efficiently presented in next chapters.

In Chapter 4, we studied the influence cascade in social networks and introduced the Heat Conduction (HC) Model which captures both social influence and non-social

influence, and extends many of the existing non-progressive models. We also presented a scalable and provably near-optimal solution for influence maximization problem by establishing three essential properties of HC: 1) submodulairty of influence spread, 2) closed form computation for influence spread, and 3) closed form greedy selection. We are the first to present a scalable solution for influence maximization under non-progressive LT model, as a special case of the HC model. We conducted extensive experiments on networks with hundreds of thousands of nodes and close to one million edges where our proposed method runs in a few minutes, in sharp contrast with the long running time of existing methods. The experiments also certified that our method outperforms the state-of-the-art in terms of both influence spread and scalability.

Chapter 5 was specified to study failure cascade in inter-dependent networks where we considered the effects of cascading failures both *within* and *across* different layers. The goal of the study was to investigate how different *couplings* (i.e., inter-dependencies) between network elements across layers affect the cascading failure dynamics. Through experiments using the proposed framework, we showed that under the one-to-one coupling map, how nodes from two inter-dependent networks are coupled together play a crucial role in the final size of the resulting failure cascades: coupling corresponding nodes from two networks with equal importance (i.e., "high-to-high" coupling) result in smaller failure cascades than other forms of inter-dependence coupling such as "random" or "high-low" coupling. In particular, given a two-layered system with two identical networks, "high-to-high" coupling produces a *mirror* effect in that the coupling exactly mirrors the cascade within each layer and does not produce additional failures than when the two networks are independent. Our results shed lights on potential strategies for mitigating cascading failures in inter-dependent networks.

In Chapter 6, we developed an oracle to answer dynamic reachabilities efficiently for failure (deletion) prone networks (and not insertion) with frequent reahcability query requirement. In contrast to state-of-the-art which require an update after any changes in the network to answer the queries, our method does not require any update, once it is computed in $O(n^{\omega})$, if the size of failures remain in $O(1)$ compared with the network size n. Moreover, the query time for our method is $O(1)$ which is not the case for majority of the art. We also extended the definition of articulation points to the directed networks and provided formulation to find the articulation points of a network. Introducing

a related metric, called load balancing, and conducting experiments on several real networks and generative models, we showed that random network followed by Italian power grid reflect the highest load balancing across their nodes.

In Chapter 6, we also developed the pivotality metric for assessing pivotality of nodes in the reachability of a source node to a target node. Intuitively, high pivotal nodes are the ones that make the reachability occur in shorter distance as they are traversed by a large of (shorter) paths for the reachability of source to target. Using some simple network examples, we compare our pivotality metric - the avoidance-transit hitting time (ATH) metric - with other metrics defined using the shortest paths, maximum flow, and classical hitting time methods and demonstrated that these existing metrics fail to properly capture and assess the pivotality of nodes in the reachability from a source to a target while our ATH metric can. Finally, we applied the ATH method to two real-world network examples to rank the nodes based on their pivotality for the reachability from a source to a target. We visualized the results to demonstrate the performance of our proposed method.

In Chapter 7, we developed a generative model to generate a continuum from shortest-path routing to all-path routing which provides both a closed form formulation for computing the continuum distances and an efficient routing strategy. We showed that our model is generalizable for supporting multiple targets and , in addition, it builds a unifying framework for network measure computations such as centrality measures, distance measures, and topological index.

In Chapter 8, we devised an oracle for answering SSSP queries efficiently with query time of $O(m)$, space requirement of $O(n^2)$, and pre-processing time of $O(n^\omega)$, where ω is the exponent of fast matrix multiplication and currently is equal to 2.376. For this purpose, we derived the required bound for the evaporating parameter α, from the continuum method in Chapter 7, and developed a shortest path routing strategy accordingly. We proved that the bound is tight for a special designed network, but through extensive experiments over 55 real networks and generative models, we showed that the required α is much more relaxed. We also proposed and trained a machine learning method (a boosted decision tree) to learn the required α for each inputted network based on 20 network local features.

In Chapter 9, we proved that the same theory developed in Chapter 8 can be exploited

to answer replacement path queries in the case of multiple failures which was considered an open problem. For a fast query time, we leveraged from the incremental computation of fundamental matrix, the theorem we had developed in Theory chapters.

References

[1] D. Acemoglu, G. Egorov, and K. Sonin. Political model of social evolution. *Proceedings of the National Academy of Sciences*, 108(Supplement 4):21292–21296, 2011.

[2] L. A. Adamic and N. Glance. The political blogosphere and the 2004 U.S. election: Divided they blog. In *LinkKDD*, pages 36 – 43, 2005.

[3] L. A. Adamic and N. Glance. The political blogosphere and the 2004 us election: divided they blog. In *Proceedings of the 3rd international workshop on Link discovery*, pages 36–43. ACM, 2005.

[4] A. Agarwal and J. Lang. *Foundations of analog & digital electronic circuits*. 2005.

[5] M. Al-Fares, A. Loukissas, and A. Vahdat. A scalable, commodity data center network architecture. In *SIGCOMM*. ACM New York, NY, USA, 2008.

[6] M. Alamgir and U. V. Luxburg. Phase transition in the family of p-resistances. In *Advances in Neural Information Processing Systems*, pages 379–387, 2011.

[7] R. Albert, H. Jeong, and A.-L. Barabási. Error and attack tolerance of complex networks. *Nature*, 406(6794):378–382, 2000.

[8] D. Aldous and J. A. Fill. *Reversible Markov chains and random walks on graphs*. 2002.

[9] G. D. Bader and C. W. Hogue. An automated method for finding molecular complexes in large protein interaction networks. *BMC bioinformatics*, 4(1):2, 2003.

[10] P. Bak, C. Tang, and K. Wiesenfeld. Self-organized criticality. *Physical review A*, 38(1):364, 1988.

[11] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

[12] A.-L. Barabâsi, H. Jeong, Z. Néda, E. Ravasz, A. Schubert, and T. Vicsek. Evolution of the social network of scientific collaborations. *Physica A: Statistical mechanics and its applications*, 311(3):590–614, 2002.

[13] R. Bellman. On a routing problem. *Quarterly of applied mathematics*, 16(1):87–90, 1958.

[14] E. Bendito, A. Carmona, A. Encinas, J. Gesto, and M. Mitjana. Kirchhoff indexes of a network. *Linear algebra and its applications*, 432(9):2278–2292, 2010.

[15] A. Bernstein and D. Karger. Improved distance sensitivity oracles via random sampling. In *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 34–43. Society for Industrial and Applied Mathematics, 2008.

[16] A. Bernstein and D. Karger. A nearly optimal oracle for avoiding failed vertices and edges. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 101–110. ACM, 2009.

[17] S. Biswas and R. Morris. Exor: opportunistic multi-hop routing for wireless networks. *ACM SIGCOMM Computer Communication Review*, 35(4):133–144, 2005.

[18] D. Boley, G. Ranjan, and Z.-L. Zhang. Commute times for a directed graph using an asymmetric laplacian. *Linear Algebra and its Applications*, 435(2):224–242, 2011.

[19] S. P. Borgatti and M. G. Everett. A graph-theoretic perspective on centrality. *Social networks*, 28(4):466–484, 2006.

[20] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. 1998.

[21] C. D. Brummitt, R. M. DâĂŹSouza, and E. Leicht. Suppressing cascades of load in interdependent networks. *Proceedings of the National Academy of Sciences*, 109(12):E680–E689, 2012.

[22] S. V. Buldyrev, R. Parshani, G. Paul, H. E. Stanley, and S. Havlin. Catastrophic cascade of failures in interdependent networks. *Nature*, 464(7291):1025–1028, 2010.

[23] L. Calvet and A. Fisher. Forecasting multifractal volatility. *Journal of econometrics*, 105(1):27–58, 2001.

[24] P. J. Carrington, J. Scott, and S. Wasserman. *Models and methods in social network analysis*, volume 28. Cambridge university press, 2005.

[25] M. Cha, A. Mislove, and K. P. Gummadi. A measurement-driven analysis of information propagation in the flickr social network. In *WWW*, pages 721 – 730, 2009.

[26] D. Chamberlin. Xquery: An xml query language. *IBM systems journal*, 41(4):597–615, 2002.

[27] P. Chebotarev. A class of graph-geodetic distances generalizing the shortest-path and the resistance distances. *Discrete Applied Mathematics*, 159(5):295–302, 2011.

[28] S. Chechik, M. Langberg, D. Peleg, and L. Roditty. f-sensitivity distance oracles and routing schemes. *Algorithmica*, 63(4):861–882, 2012.

[29] M. Chen, J. Liu, and X. Tang. Clustering via random walk hitting time on directed graphs. In *AAAI*, volume 8, pages 616–621, 2008.

[30] W. Chen, C. Wang, and Y. Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *KDD*, pages 1029 – 1038, 2010.

[31] W. Chen, Y. Yuan, and L. Zhang. Scalable influence maximization in social networks under the linear threshold model. In *ICDM*, pages 88 – 97, 2010.

[32] Y. Chen and Y. Chen. An efficient algorithm for answering graph reachability queries. In *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*, pages 893–902. IEEE, 2008.

[33] A. Clauset, C. Moore, and M. E. J. Newman. Hierarchical structure and the prediction of missing links in networks. *Nature*, 453:98–101, 2008.

[34] P. Clifford and A. Sudbury. A model for spatial conflict. *Biometrika*, 60(3):581 – 588, Dec. 1973.

[35] D. Coppersmith. Rectangular matrix multiplication revisited. *Journal of Complexity*, 13(1):42–49, 1997.

[36] H. Corley and Y. S. David. Most vital links and nodes in weighted networks. *Operations Research Letters*, 1(4):157–160, 1982.

[37] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, et al. *Introduction to algorithms*, volume 2. MIT press Cambridge, 2001.

[38] C. Demetrescu and G. F. Italiano. Fully dynamic transitive closure: breaking through the o (n/sup 2/) barrier. In *Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on*, pages 381–389. IEEE, 2000.

[39] C. Demetrescu, M. Thorup, R. A. Chowdhury, and V. Ramachandran. Oracles for distances avoiding a failed node or link. *SIAM Journal on Computing*, 37(5):1299–1318, 2008.

[40] P. Domingos and M. Richardson. Mining the network value of customers. In *KDD*, pages 57 – 66, 2001.

[41] P. G. Doyle and J. L. Snell. *Random walks and electric networks*. 1984.

[42] R. Duan and S. Pettie. Dual-failure distance and connectivity oracles. In *Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 506–515. Society for Industrial and Applied Mathematics, 2009.

[43] N. Due, L. Song, M. G. Rodriguez, and H. Zha. Scalable influence estimation in continuous-time diffusion networks. In *NIPS*, 2013.

[44] D. Eppstein. Finding the k shortest paths. *SIAM Journal on computing*, 28(2):652–673, 1998.

[45] P. Erdös and A. Rényi. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*, 5(17-61):43, 1960.

[46] P. ErdÅŚs and A. RÃľnyi. On the evolution of random graphs. In *PUBLICATION OF THE MATHEMATICAL INSTITUTE OF THE HUNGARIAN ACADEMY OF SCIENCES*, pages 17– 61, 1960.

[47] ESNet. Us energy science network. `http://www.es.net/`.

[48] E. Even-Dar and A. Shapira. A note on maximizing the spread of influence in social networks. In *WINE*, pages 281 – 286, 2007.

[49] L. R. Ford Jr. Network flow theory. Technical report, RAND CORP SANTA MONICA CA, 1956.

[50] F. Fouss, A. Pirotte, J.-M. Renders, and M. Saerens. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *IEEE Transactions on knowledge and data engineering*, 19(3):355–369, 2007.

[51] K. Françoisse, I. Kivimäki, A. Mantrach, F. Rossi, and M. Saerens. A bag-of-paths framework for network data analysis. *Neural Networks*, 90:90–111, 2017.

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

[53] L. C. Freeman. The gatekeeper, pair-dependency and structural centrality. *Quality & Quantity*, 14(4):585–592, 1980.

[54] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin. Highly-resilient, energy-efficient multipath routing in wireless sensor networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(4):11–25, 2001.

[55] F. R. Gantmacher. *Theory of Matrices. 2V.* Chelsea publishing company, 1960.

[56] G. Golnari, A. Asiaee, A. Banerjee, and Z.-L. Zhang. Revisiting non-progressive influence models: Scalable influence maximization in social networks. In *UAI*, pages 316–325, 2015.

[57] G. Golnari, A. Asiaee T., A. Banerjee, and Z.-L. Zhang. Revisiting non-progressive influence models: scalable influence maximization. *arXiv:1412.5718*, 2014.

[58] G. Golnari and D. Boley. Continuum of all-pair shortest-path to all-path via random walk. 2013.

[59] G. Golnari, D. Boley, and Z.-L. Zhang. Advanced random walk metrics and their application in pivotality measures, routing continuum, and replacement path problem. *to be submitted to Linear Algebra and its Applications.*

[60] G. Golnari, D. Boley, and Z.-L. Zhang. Random walk fundamental tensor and network analysis. *submitted to Linear Algebra and its Applications.*

[61] G. Golnari, Y. Li, and Z.-L. Zhang. Pivotality of nodes in reachability problems using avoidance and transit hitting time metrics. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1073–1078. ACM, 2015.

[62] G. Golnari and Z.-L. Zhang. The effect of different couplings on mitigating failure cascades in interdependent networks. In *Computer Communications Workshops (INFOCOM WKSHPS), 2015 IEEE Conference on*, pages 677–682. IEEE, 2015.

[63] M. Gomez-Rodriguez, D. Balduzzi, and B. SchÃűlkopf. Uncovering the temporal dynamics of diffusion networks. In *ICML*, 2011.

[64] M. Gomez-Rodriguez and B. Scholkopf. Influence maximization in continuous time diffusion networks. In *ICML*, 2012.

[65] A. Goyal, W. Lu, and L. V. Lakshmanan. Simpath: An efficient algorithm for influence maximization under the linear threshold model. In *ICDM*, pages 211–220, 2011.

[66] M. R. Henzinger and V. King. Fully dynamic biconnectivity and transitive closure. In *Foundations of Computer Science, 1995. Proceedings., 36th Annual Symposium on*, pages 664–672. IEEE, 1995.

[67] J. Hershberger and S. Suri. Vickrey prices and shortest paths: What is an edge worth? In *Foundations of Computer Science, 2001. Proceedings. 42nd IEEE Symposium on*, pages 252–259. IEEE, 2001.

[68] R. Holley and T. Liggett. Ergodic theorems for weakly interacting infinite systems and the voter model. *The Annals of Probability*, 3(4):643 – 663, 1975.

[69] F. P. Incropera. *Fundamentals of heat and mass transfer*. John Wiley & Sons, 2011.

[70] G. F. Italiano, L. Laura, and F. Santaroni. Finding strong bridges and strong articulation points in linear time. *Theoretical Computer Science*, 447:74–84, 2012.

[71] R. Jin and G. Wang. Simple, fast, and scalable reachability oracle. *Proceedings of the VLDB Endowment*, 6(14):1978–1989, 2013.

[72] J. G. Kemeny, J. L. Snell, et al. *Finite markov chains*, volume 356. van Nostrand Princeton, NJ, 1960.

[73] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146. ACM, 2003.

[74] D. Kempe, J. Kleinberg, and v. Tardos. Maximizing the spread of influence through a social network. In *KDD*, pages 137 – 146, 2003.

[75] A. R. Khakpour and A. X. Liu. Quantifying and querying network reachability. In *Distributed Computing Systems (ICDCS), 2010 IEEE 30th International Conference on*, pages 817–826. IEEE, 2010.

[76] N. Khanna and S. Baswana. Approximate shortest paths avoiding a failed vertex: Optimal size data structures for unweighted graph. In *27th International Symposium on Theoretical Aspects of Computer Science-STACS 2010*, pages 513–524, 2010.

[77] M. Kimura and K. Saito. Tractable models for information diffusion in social networks. In *PKDD*, pages 259–271. 2006.

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

[79] L. Kleinrock. *Queueing systems, volume 2: Computer applications*, volume 66. wiley New York, 1976.

[80] V. Kopp, V. Kaganer, J. Schwarzkopf, F. Waidick, T. Remmele, A. Kwasniewski, and M. Schmidbauer. X-ray diffraction from nonperiodic layered structures with correlations: analytical calculation and experiment on mixed aurivillius films. *Acta Crystallographica Section A: Foundations of Crystallography*, 68(1):148–155, 2012.

[81] P. S. Kutchukian, D. Lou, and E. I. Shakhnovich. Fog: Fragment optimized growth algorithm for the de novo generation of molecules occupying druglike chemical space. *Journal of chemical information and modeling*, 49(7):1630–1642, 2009.

[82] G. F. Lawler. *Random walk and the heat equation*. 2010.

[83] C. E. Leiserson. Fat-trees: universal networks for hardware-efficient supercomputing. *IEEE transactions on Computers*, 100(10):892–901, 1985.

[84] C. E. Leiserson. Fat-trees: universal networks for hardware-efficient supercomputing. *Computers, IEEE Transactions on*, 100(10):892–901, 1985.

[85] J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, and Z. Ghahramani. Kronecker graphs: An approach to modeling networks. *J. Mach. Learn. Res.*, 11:985–1042, 2010.

[86] J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, and Z. Ghahramani. Kronecker graphs: An approach to modeling networks. *Journal of Machine Learning Research*, 11(Feb):985–1042, 2010.

[87] J. Leskovec, D. Huttenlocher, and J. Kleinberg. Predicting positive and negative links in online social networks. In *WWW*, pages 641 – 650, 2010.

[88] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):2, 2007.

[89] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance. Cost-effective outbreak detection in networks. In *KDD*, pages 420 – 429, 2007.

[90] J. Leskovec and J. J. Mcauley. Learning to discover social circles in ego networks. In *Advances in neural information processing systems*, pages 539–547, 2012.

[91] Y. Li and Z.-L. Zhang. Random walks on digraphs: A theoretical framework for estimating transmission costs in wireless routing. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9. IEEE, 2010.

[92] Y. Li, Z.-L. Zhang, and D. Boley. The routing continuum from shortest-path to all-path: A unifying theory. In *Distributed Computing Systems (ICDCS), 2011 31st International Conference on*, pages 847–856. IEEE, 2011.

[93] F. Merz and P. Sanders. Preach: A fast lightweight reachability index using pruning and contraction hierarchies. In *European Symposium on Algorithms*, pages 701–712. Springer, 2014.

[94] M. M. Michael. Safe memory reclamation for dynamic lock-free objects using atomic reads and writes. In *Proceedings of the twenty-first annual symposium on Principles of distributed computing*, pages 21–30. ACM, 2002.

[95] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functionsâĂŤI. *Mathematical Programming*, 14(1):265 – 294, Dec. 1978.

[96] E. D. Nering. Linear algebra and matrix theory. Technical report, 1970.

[97] M. E. Newman. Scientific collaboration networks. i. network construction and fundamental results. *Physical review E*, 64(1):016131, 2001.

[98] M. E. Newman. Assortative mixing in networks. *Physical review letters*, 89(20):208701, 2002.

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

[100] D. T. Nguyen, Y. Shen, and M. T. Thai. Detecting critical nodes in interdependent power networks for vulnerability assessment. *IEEE Transactions on Smart Grid (ToSG)*, 4(1):151–159, March 2013. Special Issues on *Smart Grid Communication Systems: Reliability, Dependability & Performance*.

[101] J. D. Noh and H. Rieger. Random walks on complex networks. *Physical review letters*, 92(11):118701, 2004.

[102] J. R. Norris. *Markov chains*. Number 2. Cambridge university press, 1998.

[103] J. L. Palacios and J. M. Renom. Bounds for the kirchhoff index of regular graphs via the spectra of their random walks. *International Journal of Quantum Chemistry*, 110(9):1637–1641, 2010.

[104] M. Parandehgheibi and E. Modiano. Robustness of interdependent networks: The case of communication networks and the power grid. In *Global Communications Conference (GLOBECOM), 2013 IEEE*, pages 2164–2169.

[105] R. Parshani, S. V. Buldyrev, and S. Havlin. Interdependent networks: Reducing the coupling strength leads to a change from a first to second order percolation transition. *Physical review letters*, 105(4):048701, 2010.

[106] N. Pathak, A. Banerjee, and J. Srivastava. A generalized linear threshold model for multiple cascades. In *ICDM*, pages 965 – 970, 2010.

[107] S. Pettie. A new approach to all-pairs shortest paths on real-weighted graphs. *Theoretical Computer Science*, 312(1):47–74, 2004.

[108] L. Popa, C. Raiciu, I. Stoica, and D. S. Rosenblum. Reducing congestion effects in wireless networks by multipath routing. In *Network Protocols, 2006. ICNP'06. Proceedings of the 2006 14th IEEE International Conference on*, pages 96–105. IEEE, 2006.

[109] G. Ranjan and Z.-L. Zhang. How to "glue" a robust smart-grid: A finite network theory of inter-dependent networks. In 7th Cyber Security & Information Intelligence Research Workshop, Sep 2011.

[110] G. Ranjan and Z.-L. Zhang. Geometry of complex networks and topological centrality. *Physica A: Statistical Mechanics and its Applications*, 392(17):3833–3845, 2013.

[111] E. Ravasz, A. L. Somera, D. A. Mongru, Z. N. Oltvai, and A.-L. Barabási. Hierarchical organization of modularity in metabolic networks. *science*, 297(5586):1551–1555, 2002.

[112] L. Roditty. A faster and simpler fully dynamic transitive closure. In *Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 404–412. Society for Industrial and Applied Mathematics, 2003.

[113] L. Roditty and U. Zwick. A fully dynamic reachability algorithm for directed graphs with an almost linear update time. *SIAM Journal on Computing*, 45(3):712–733, 2016.

[114] V. Rosato, L. Issacharoff, F. Tiriticco, S. Meloni, S. Porcellinis, and R. Setola. Modelling interdependent infrastructures using interacting dynamical models. *International Journal of Critical Infrastructures*, 4(1-2):63–79, 2008.

[115] V. Rosato, L. Issacharoff, F. Tiriticco, S. Meloni, S. Porcellinis, and R. Setola. Modelling interdependent infrastructures using interacting dynamical models. *International Journal of Critical Infrastructures*, 4(1):63–79, 2008.

[116] D. H. Rouvray. The role of the topological distance matrix in chemistry. Technical report, DTIC Document, 1985.

[117] D. H. Rouvray. Predicting chemistry from topology. *Scientific American*, 255(3):40, 1986.

[118] C. M. Schneider, N. Yazdani, N. A. Araújo, S. Havlin, and H. J. Herrmann. Towards designing robust coupled networks. *Scientific reports*, 3, 2013.

[119] J. Sherman and W. J. Morrison. Adjustment of an inverse matrix corresponding to a change in one element of a given matrix. *The Annals of Mathematical Statistics*, 21(1):124–127, 1950.

[120] P. Snell and P. Doyle. Random walks and electric networks. *Free Software Foundation*, 2000.

[121] A. Stojmirović and Y.-K. Yu. Information flow in interaction networks. *Journal of Computational Biology*, 14(8):1115–1143, 2007.

[122] V. Strassen. Gaussian elimination is not optimal. *Numerische Mathematik*, 13(4):354–356, 1969.

[123] A. Tahbaz-Salehi and A. Jadbabaie. A one-parameter family of distributed consensus algorithms with boundary: From shortest paths to mean hitting times. In *Decision and Control, 2006 45th IEEE Conference on*, pages 4664–4669. IEEE, 2006.

[124] J. Tang, D. Zhang, and L. Yao. Social network extraction of academic researchers. In *ICDM*, pages 292–301, 2007.

[125] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su. ArnetMiner: extraction and mining of academic social networks. In *KDD*, pages 990–998, 2008.

[126] P. Tetali. Random walks and the effective resistance of networks. *Journal of Theoretical Probability*, 4(1):101–109, 1991.

[127] M. Thorup and U. Zwick. Compact routing schemes. In *Proceedings of the thirteenth annual ACM symposium on Parallel algorithms and architectures*, pages 1–10. ACM, 2001.

[128] M. Thorup and U. Zwick. Approximate distance oracles. *Journal of the ACM (JACM)*, 52(1):1–24, 2005.

[129] J. Van Helden, A. Naim, R. Mancuso, M. Eldridge, L. Wernisch, D. Gilbert, and S. J. Wodak. Representing and analysing molecular and cellular function in the computer. *Biological chemistry*, 381(9-10):921–935, 2000.

[130] U. Von Luxburg, A. Radl, and M. Hein. Hitting and commute times in large random neighborhood graphs. *Journal of Machine Learning Research*, 15(1):1751–1798, 2014.

[131] H. Wang, H. He, J. Yang, P. S. Yu, and J. X. Yu. Dual labeling: Answering graph reachability queries in constant time. In *Data Engineering, 2006. ICDE'06. Proceedings of the 22nd International Conference on*, pages 75–75. IEEE, 2006.

[132] D. J. Watts and S. H. Strogatz. Collective dynamics of'small-world'networks. *nature*, 393(6684):440, 1998.

[133] O. Weimann and R. Yuster. Replacement paths via fast matrix multiplication. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pages 655–662. IEEE, 2010.

[134] O. Weimann and R. Yuster. Replacement paths and distance sensitivity oracles via fast matrix multiplication. *ACM Transactions on Algorithms (TALG)*, 9(2):14, 2013.

[135] H. Wiener. Structural determination of paraffin boiling points. *Journal of the American Chemical Society*, 69(1):17–20, 1947.

[136] K. G. Wilson. Quantum chromodynamics on a lattice. In *New Developments in Quantum Field Theory and Statistical Mechanics Cargèse 1976*, pages 143–172. Springer, 1977.

[137] G. G. Xie, J. Zhan, D. A. Maltz, H. Zhang, A. Greenberg, G. Hjalmtysson, and J. Rexford. On static reachability analysis of ip networks. In *INFOCOM 2005*, volume 3, pages 2170–2183. IEEE, 2005.

[138] R. Yuster and U. Zwick. Answering distance queries in directed graphs using fast matrix multiplication. In *Foundations of Computer Science, 2005. FOCS 2005. 46th Annual IEEE Symposium on*, pages 389–396. IEEE, 2005.

[139] W. Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33:452 – 473, 1977.

[140] B. Zhou and N. Trinajstić. A note on kirchhoff index. *Chemical Physics Letters*, 455(1):120–123, 2008.