CHANGING EDGES IN GRAPHICAL MODEL ALGORITHMS

BY

LINJIA CHANG

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2016

Urbana, Illinois

Adviser:

Assistant Professor Lav R. Varshney

ABSTRACT

Graphical models are used to describe the interactions in structures, such as the nodes in decoding circuits, agents in small-world networks, and neurons in our brains. These structures are often not static and can change over time, resulting in removal of edges, extra nodes, or changes in weights of the links in the graphs. For example, wires in message-passing decoding circuits can be misconnected due to process variation in nanoscale manufacturing or circuit aging, the style of passes among soccer players can change based on the team's strategy, and the connections among neurons can be broken due to Alzheimer's disease. The effects of these changes in graphs can reveal useful information and inspire approaches to understand some challenging problems.

In this work, we investigate the dynamic changes of edges in graphs and develop mathematical tools to analyze the effects of these changes by embedding the graphical models in two applications.

The first half of the work is about the performance of message-passing LDPC decoders in the presence of permanently and transiently missing connections, which is equivalent to the removal of edges in the codes' graphical representation Tanner graphs. We prove concentration and convergence theorems that validate the use of density evolution performance analysis and conclude that arbitrarily small error probability is not possible for decoders with missing connections. However, we find suitably defined decoding thresholds for communication systems with binary erasure channels under peeling decoding, as well as binary symmetric channels under Gallager A and B decoding. We see that decoding is robust to missing wires, as decoding thresholds degrade smoothly. Surprisingly, we discovered the stochastic facilitation (SF) phenomenon in Gallager B decoders where having more missing connections helps improve the decoding thresholds under some conditions.

The second half of the work is about the advantages of the semi-metric property of complex weighted networks. Nodes in graphs represent elements in systems

and edges describe the level of interactions among the nodes. A semi-metric edge in a graph, which violates the triangle inequality, indicates that there is another latent relation between the pair of nodes connected by the edge. We show the equivalence between modelling a sporting event using a stochastic Markov chain and an algebraic diffusion process, and we also show that using the algebraic representation to calculate the stationary distribution of a network can preserve the graph's semi-metric property, which is lost in stochastic models. These semi-metric edges can be treated as redundancy and be pruned in the all-pairs shortest-path problems to accelerate computations, which can be applied to more complicated problems such as PageRank. We then further demonstrate the advantages of semi-metricity in graphs by showing that the percentage of semi-metric edges in the interaction graphs of two soccer teams changes linearly with the final score. Interestingly, these redundant edges can be interpreted as a measure of a team's tactics.

To my family, for their love and support.

ACKNOWLEDGMENTS

First I would like to thank my adviser Prof. Lav R. Varshney for his patience, support, and advice through this work and over the past two years. He is not only an excellent thesis adviser, but also a mentor and a friend. I also thank my group mate Dr. Avhishek Chatterjee, who jointly developed the work on LDPC decoders and shared his insights. The work on complex weighted networks in Chapter 4 was suggested and jointly developed by Dr. Tiago Simas, who supervised me at my internship at Telefónica R&D. I thank him for providing guidance and encouraging me to pursue this direction. I would also like to thank my dear group mates Ravi Kiran and Daewon Seo, office mates, graduate students in the SINE group in the Coordinated Science Laboratory, and friends in the Electrical and Computer Engineering department, for teaching me new things, discussing ideas, and encouraging me to overcome the difficulties. I am extremely grateful for the opportunity to learn and grow at this great institution. Lastly, I thank my parents and family for their limitless love and support, without which I would not be the person I am today.

- This work was supported in part by Systems on Nanoscale Information fabriCs (SONIC), one of the six SRC STARnet Centers, sponsored by MARCO and DARPA.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Graph models are everywhere and they have been used in a wide range of applications such as coding theory, signal processing, and machine learning. As real systems change over time, dynamic changes happen to their corresponding graph models. Nodes or edges can be removed from or added to the graphs, an edge can connect a different pair of nodes, and the weights of the existing edges in the graphs can also vary. It is of interest for us to answer the questions: what are the effects of changing edges in a graph and how can we analyze these changes systematically?

In this thesis, we aim to shine light on this topic by looking at two cases: defective decoders for low-density parity-check (LDPC) codes and algebraic representations of stochastic Markov chains in networks. The two examples represent graphs with missing and redundant edges, respectively.

LDPC, expander, and spatially coupled codes are all codes that can be represented by graphs, and LDPC codes especially, have been the focus of a great deal of attention and are now used in a variety of applications such as satellite communications, 10GBase-T Ethernet, the Wi-Fi IEEE 802.11 standard, and storage systems. The bipartite Tanner graph is not only an elegant graphical representation of LDPC codes, but is also how the decoding circuits are designed and built. In iterative message-passing decoders, messages are passed along the edges in the graph and the computation is performed at the nodes in order to correct corrupted messages received from the channel.

With the rapid development of nanoscale circuit manufacturing, circuits are now built with shrinking device area, higher component density, lower power-supply voltage, and more critical timing constraints. This brings rising concerns that the computation elements, such as gates, wires, switches, etc., can be unreliable and may produce errors. Varshney [1] has characterized LDPC decoders with noisy computation elements. In the first half of this work, instead of noisy decoders, we study the performance of LDPC decoders with permanent and tran-

sient wiring errors, where the connection graphs of the decoder architectures are different from the designed patterns. In particular, we focus on decoders with missing connections, where edges are taken out from the corresponding Tanner graphs.

Chapter 2 motivates the problem of LDPC decoders with missing connections in detail, and discusses models of codes, channels, and LDPC decoders with both transiently and permanently missing connections, with a particular focus on hardware modeling. It also develops tools including concentration and convergence theorems that provide validity to density evolution analysis in Sec. 2.4. Chapter 3 analyzes the peeling decoder on the binary erasure channel (BEC) and the Gallager A and Gallager B decoders on binary symmetric channel (BSC) using density evolution, characterizing P_e with missing connections. In Ch. 4, we perform sensitivity analysis of density evolution to give insight into whether manufacturing or operational resources are more important in communication infrastructures. We also comment on how our results inform semiconductor manufacturing yield analysis under the new paradigm of allowing some level of wiring error. To demonstrate the practical utility of density evolution analysis, we also perform finite-length simulations of decoders with missing connections.

While the first half of the thesis studies the case of missing edges in graphs, the second half of the work studies the case of redundant edges in graphs. We live in the age of computing where many systems, including sporting events and the internet, can be analyzed using data and algorithms developed based on mathematical models. Researchers have achieved great successes in anomaly detection, information spread prediction, and diffusion models based on stochastic Markov chains in general. Weighted graphs, where the weights of edges can be considered as beliefs, have been well studied in the field of fuzzy set theory. The focus has been on the mathematics of weighted graphs and their algebraic characteristics. Simas and Mocha [2] connected the field of fuzzy set theory with complex networks by developing methods to compute the transitive closure of weighted graphs obtained from data. Here, we extend their work and establish the equivalence between stochastic diffusion models based on Markov chains and the algebraic transitive closure of weighted graphs. The semi-metric edges, often treated as redundant connections in distance graphs, enable us to identify indirectly related elements in networks, which can benefit applications such as community detection, sports analytics, recipe recommendation, and so on. Using the algebraic method allows us to preserve the semi-metric property of graphs, resulting

in more effective information extraction from data.

Chapter 5 introduces the background and establishes the connection between stochastic diffusion model and its algebraic belief network. Chapter 6 demonstrates the advantages of the algebraic methods with two examples: predicting goals in soccer games and calculating the Internet PageRank. A more detailed outline is given in the introductory section of Ch. 5.

Bibliographical Note

The problem of LDPC decoders and part of the results of Ch. 2–4 have been presented at the conference listed below and appear in its proceedings:

- L. Chang, A. Chatterjee, and L. R. Varshney, "LDPC decoders with missing connections," in *Proceedings of the 2016 IEEE International Symposium on Information Theory*, Barcelona, Spain, 10–15 July 2016, pp.1576–1580.

and also appear in

- L. Chang, A. Chatterjee, and L. R. Varshney, "Performance of LDPC decoders with missing connections," *IEEE Transactions on Communications*, to appear.

The problem formulation and results of algebraic representation of stochastic Markov chain in Ch. 5 and 6 appear in the manuscript

- T. Simas, L. Chang, A. Bassolas, A. Diaz-Guilera, P. Obrador, and P. Rodriguez, "Algebraic Representation of Stochastic Markov Chain."

CHAPTER 2

DEFECTIVE LDPC DECODERS

Low-density parity-check (LDPC) codes are prevalent due to their performance near the Shannon limit with message-passing decoders that have efficient implementation [3]. With the end of CMOS scaling nearing, there is interest in nanoscale circuit implementations of decoders, but this introduces concerns that process variation in manufacturing may lead to interconnect patterns different than designed [4–6], especially under self-assembly [7, 8]. Yield on manufactured chips deemed perfectly operational is small—reports indicate 1–15% of circuit elements such as wires, switches, and transistors are defective [8]—leading to rather expensive industrial waste [9]. Changing the paradigm of circuit functionality from perfection to some small probability α of missing wires may eliminate much wastage and so it is of interest to characterize chips with permanently missing connections to determine suitable error tolerances.

Process variation in manufacturing also causes fluctuation in device geometries, which might prevent them from meeting timing constraints [10], especially in future nanoscale technologies like carbon nanotube circuits where device geometry control is especially difficult. Such timing errors lead to missed messages in intra-chip communications, equivalent to transiently missing connections. Connections can also be missing transiently in programmable LDPC decoders [11]. It is thus also of interest to characterize decoders with transiently missing connections.

However, most fault-tolerant computing research assumes the circuit is constructed correctly and is concerned only with faults in computational elements. Peter Elias noted the following [12], but it remains true today:

> J. Von Neumann has analyzed computers whose unreliable elements are majority organs—crude models of a neuron. Shannon and Moore have analyzed combinational circuits whose components are unreliable relays. Both papers assume that the wiring diagram is correctly drawn and correctly followed in construction, but that computation proper is performed only by unreliable elements.

Such assumptions of fault-free circuit construction need to be reevaluated and performance analysis of computation with such wiring faults needs to be carried out. The only work we are aware of in fault-tolerant computing theory that briefly discusses wiring errors is the monograph of Winograd and Cowan [13, Ch. 9.2].

Varshney [1] had previously extended the method of density evolution to decoders with faults in the computational elements and showed that it is possible to communicate with arbitrarily small error probability with noisy Gaussian belief propagation. Asymptotic characterizations were also determined for Gallager A [1] and Gallager B decoders with transient noise [14–16], energy optimization [17], and both permanent and transient noise [18]. Noisy decoding [19–25], and general noisy belief propagation, not necessarily in decoding [26, 27], have also been studied. Recent studies show that bit-flipping decoders with data-dependent gate failures can achieve zero error probability [25, 28], but with a subset of computation hardware that is reliable and no wiring diagram errors.

Rather than noise in computational elements, here we analyze the performance of message-passing decoders with missing connections and show that appropriately defined decoding thresholds are robust, in the sense of degrading smoothly. This is true for both transiently and permanently missing connections in message-passing decoding circuits. In certain settings, missing connections actually improve performance, resulting in stochastic facilitation (SF).[1]

A key difference between noisy computational elements and missing connections is that circuit technology enables detection of missing connections (Sec. 2.3). This allows for simple adaptations of decoding algorithms, yielding better decoding performance under missing connections than under noisy components. A notable manifestation of this difference is in the so-called decoder useful region. For transient or permanent noise, there is a strictly positive lower bound for the useful region, below which the channel output is actually better than the decoded version since the internal decoder noise makes things worse. For missing connections, there is no such lower boundary since the decoder asymptotically never degrades performance from the raw channel error rate.

The celebrated results of Richardson and Urbanke [31] developed density evolution for analyzing message-passing decoders for LDPC codes that are correctly wired. Here we extend those results, so we can use the density evolution technique to characterize symbol error rate P_e, measuring the fraction of incorrectly

[1] SF in decoding was observed with transient errors in computation, rather than with missing connections, initially in memory recall [27, 29] and then in communications [23, 30].

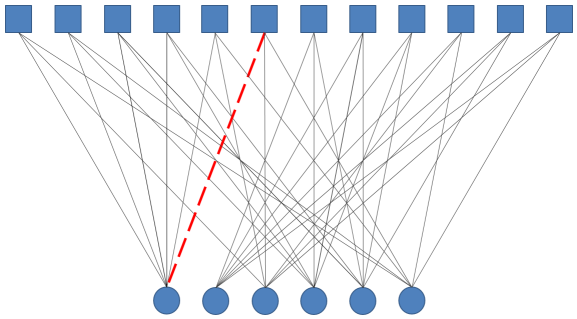


Figure 2.1: Tanner graph of a $(3, 6)$ regular LDPC code, with a missing wire for a corresponding message-passing decoder highlighted with a dashed line.

decoded symbols at the end of message-passing decoding, even when the decoder has missing connections. We also show that the performance of decoders with transiently and with permanently missing connections are asymptotically equivalent. Traditionally [31], there are thresholds for channel noise level ε below which P_e can be driven to 0 with increasing blocklength n. Unfortunately with missing connections in message-passing decoders, P_e cannot be driven to 0 in general without a significant modification of the decoding algorithm. Thus, following [1], we let η be an upper bound to the final error probability that can be achieved by decoders with missing connections after many iterations ℓ and give thresholds to ε, below which $\lim_{\ell\to\infty} P_e^{(\ell)} \leq \eta$ under density evolution.

In the following sections, We define the code and channels considered in this work and construct fault-free and missing-wire decoder models for characterization later. We then develop mathematical tools to analyze the performances of LDPC decoders.

2.1 Ensemble of LDPC Codes and Channels

We are concerned with the standard LDPC code ensemble G^n, both regular and irregular. First consider the ensemble $G^n(d_\mathsf{v}, d_\mathsf{c})$ of regular LDPC codes of length n, which can be defined by a bipartite Tanner graph with n variable nodes of degree d_v in one set, and $nd_\mathsf{v}/d_\mathsf{c}$ check nodes of degree d_c in the other set (see Fig. 2.1).

For irregular codes $G^n(\lambda, \rho)$, the degree distribution of variable and check nodes are denoted by functions $\lambda(x) = \sum_{d=2}^{\infty} \lambda_d x^{d-1}$ and $\rho(x) = \sum_{d=2}^{\infty} \rho_d x^{d-1}$, where λ_d and ρ_d specify the fraction of edges in the graph that are connected to

nodes with degree d. The variable nodes hold the codeword messages, and the check nodes enforce the constraints among variable nodes according to the code design. We consider this binary linear code ensemble as defined over the alphabet $\{\pm 1\}$. Although this section is general, for convenience, let us think of the communication channel as either BSC with output alphabet $\{\pm 1\}$ or BEC with output alphabet $\{\pm 1, ?\}$.

2.2 Fault-Free Message-Passing Decoder

The decoder operates by passing messages iteratively over the edges in the Tanner graph of the code. The implementation of such message-passing decoders in hardware follows the construction of the same Tanner graph too. We define a variable-to-check node message $u_{v\to c}$ and a check-to-variable node message $u_{c\to v}$. Message $u_{v'\to c'}$ from variable node v' to check node c' is often computed based on all incoming $u_{c\to v'}$ messages, where $c \in N(v')$ is a neighboring node of v' and $c \neq c'$. For peeling, Gallager A, and Gallager B decoders, message $v_{c'\to v'}$ from check node c' to variable node v' is the product of all incoming $u_{v\to c'}$ messages, where $v \in N(c')$ is a neighboring node of c' and $v \neq v'$.

2.3 Missing Connections Model

As discussed in Ch. 1, there are two types of missing connections: permanent missing connections caused by breaks in interconnects and transient missing connections caused mainly by timing errors in intra-chip communication due to geometry variation in circuitry. Although specific statistical characterization is not reported in the semiconductor industry, process variation in manufacturing leads to both kinds of errors and can be fairly prevalent [8].

For a given decoder circuit, permanent failure is modeled by removing each connection between variable and check nodes with probability α independently from others, before decoding starts. These connections are never active once removed. Our conversations with circuit designers suggest that when an interconnect is broken in LDPC decoders implemented in a variety of device technologies, the measured signal voltage at this open-ended wire is neither low (0–$0.3V_{dd}$) nor high ($0.7V_{dd}$–V_{dd}); it is an intermediate floating value that may vary within

the range ($0.3V_{dd}$–$0.7V_{dd}$) which can be differentiated from high and low values. Hence we model it as an erasure symbol, "?".

For the transiently missing connection setting, each connection between variable and check nodes is removed independently from others with probability α at each decoding iteration. Transiently missing connections may occur due to timing error from incorrect geometry: consider a misalignment of synchronization when one branch of signal arrives after the computation at the destination node has started, especially among those circuit implementations that do not store the last signal sample. Transiently missing connections might similarly happen in programmable LDPC decoder architectures [11]. Due to the difficulty in controlling device geometries, future carbon nanotube circuits are projected to have a significant number of these transient missing connections. Again we model as an erasure symbol, "?".

For notational convenience, let us restrict attention to decoders with messages in $\{\pm 1, ?\}$, but again concentration and convergence results demonstrated in Ch. 2.4 are general. Motivated by different concerns, [32, Example 4.86] considered erasures in decoder messages as a representation of confidence, whereas [33] considered erasures as a way to capture check node or variable node failures in belief propagation.

2.4 Performance Analysis Tools

We now present mathematical tools to simplify the performance analysis of LDPC decoders with missing connections. In particular, we establish symmetry conditions for binary codes for easy analysis, and concentration and convergence results that endow the density evolution method with significance. Such results can be applied not only to decoders with binary messages, but also to those with larger message sets.

2.4.1 Restriction to All-One Codeword

Under certain symmetry conditions of the code, the communication channel, and the message-passing decoder, the probability of error is independent of the transmitted codeword. These conditions are:

C1. **Code Symmetry:** Code is a binary linear code.

C2. **Channel Symmetry:** Channel is a binary memoryless symmetric channel [32, Def. 4.3 and 4.8].

C3. **Check Node Symmetry:** If incoming messages of a check node are multiplied by $\{b_i \in \{\pm 1\}\}$, then the computed message is multiplied by $\prod_i b_i$.

C4. **Variable Node Symmetry:** If the sign of each incoming message is flipped, the sign of the computed message is also flipped.

Proposition 1. *Under conditions C1–C4, in the presence of transiently or permanently missing connections, the probability of error of a message passing decoder is independent of the transmitted codeword.*

Proof. First consider mapping the erasure message "?", sent when a connection is missing, to 0; thus the check-to-variable and variable-to-check messages are the messages computed at check node and variable node, respectively, multiplied by either 1 (connection exists) or 0 (missing connections). It is easy to see that messages passed between check and variable nodes satisfy the respective symmetry conditions [32, Def. 4.82]. Hence, the result follows by invoking [32, Lem. 4.92]. □

In the sequel, we restrict the analysis of all models to the all-one codeword.

2.4.2 Concentration around Ensemble Average

We now show that the performance of LDPC codes decoded with missing-connection decoders stays close to the expected performance of the code ensemble for both transiently and permanently missing wires. The approach follows [31] and is based on constructing an exposure Martingale, obtaining bounded difference constants, and using Azuma's inequality.

Fix the number of decoding iterations at some finite ℓ and let Z be the number of incorrect values held among all $d_{\mathrm{v}}n$ edges at the end of the ℓth iteration for a specific choice of code, channel noise, and decoder with missing wires. Let $E[Z]$ be the expectation of Z. Theorem 1 holds for decoders with both transiently and permanently missing connections.

Theorem 1 (Concentration around Expected Value)**.** *There exists a positive constant* $\beta = \beta(d_{\mathtt{v}}, d_{\mathtt{c}}, \ell)$ *such that for any* $\varepsilon > 0$*,*

$$\Pr[|Z - E[Z]| > n d_{\mathtt{v}} \varepsilon / 2] \leq 2e^{-\beta \varepsilon^2 n}.$$

Proof. See Apps. B and C for permanent and transient missing connections, respectively. □

Recall Doob's Martingale construction from [31], and the bounded difference constants for exposing channel noise realizations and the realized code connections, together with Azuma's inequality. The main difference between the Martingale construction here and [31] is in the bounded differences due to the additional randomness from missing connections.

For permanently missing connections, one can think of the final connection graph being sampled from an ensemble of irregular random graphs with binomial degree distribution with average degrees $(1-\alpha)d_{\mathtt{c}}$ and $(1-\alpha)d_{\mathtt{v}}$, bounded by maximum degrees $d_{\mathtt{c}}$ and $d_{\mathtt{v}}$. Hence, the result follows from the result for correctly-wired irregular codes [31].

For transiently missing connections, the Martingale is constructed differently. Here instead of edges, for ℓ iterations, we sequentially expose the realization of edges at different iterations. Similar to [1] for transient noise in computational elements, the Martingale difference is bounded using the maximum number of edges over which a message can propagate in ℓ iterations, by unwrapping a computation tree.

Note that β will be smaller for transient than permanent miswiring. The theorem extends directly to irregular LDPC codes.

2.4.3 Convergence to Cycle-Free Case

We now show that the average performance of an LDPC code ensemble converges to an associated cycle-free tree structure, unwrapping a computation tree as in [31].

For an edge whose connected neighborhood with depth 2ℓ is cycle-free, let q denote the expected number of incorrect values held along this edge at the end of ℓth decoding iteration. The expectation is taken over the choice of code, the messages received from the channel, and the realization of the decoder with missing wires.

The theorems hold for both transiently and permanently missing connections.

Theorem 2 (Convergence to Cycle-Free Case)**.** *There exists a positive constant* $\gamma = \gamma(d_{\mathtt{v}}, d_{\mathtt{c}}, \ell)$ *such that for any* $\varepsilon > 0$ *and* $n > \frac{2\gamma}{\varepsilon}$*,*

$$|E[Z] - nd_{\mathtt{v}}q| < nd_{\mathtt{v}}\varepsilon/2.$$

Proof. The proof is identical to [31, Thm. 2], since introducing missing connections in a cycle-free tree structure does not change its cycle-free property. □

Theorem 3 (Concentration around Cycle-Free Case)**.** *There exist positive constants* $\beta = \beta(d_{\mathtt{v}}, d_{\mathtt{c}}, \ell)$ *and* $\gamma = \gamma(d_{\mathtt{v}}, d_{\mathtt{c}}, \ell)$ *such that for any* $\varepsilon > 0$ *and* $n > \frac{2\gamma}{\varepsilon}$*,*

$$\Pr[|Z - nd_{\mathtt{v}}q| > nd_{\mathtt{v}}\varepsilon] \leq 2e^{-\beta n\varepsilon^2}.$$

Proof. Follows directly from Thms. 1 and 2. □

This concentration result holds for all message-passing decoders with missing connections. In the sequel, we consider special cases of peeling, Gallager A, and Gallager B decoders.

2.4.4 Density Evolution

With the concentration around the cycle-free case, it is clear that the symbol error rate P_e of message-passing decoders with missing connections can be characterized with the density evolution technique. Let $P_e^{(\ell)}(g, \varepsilon, \alpha)$ be the error probability of decoding a code $g \in \boldsymbol{G}^n$, after the ℓth iteration, where ε is the channel noise parameter, and α is the decoder missing wire probability. Density evolution evaluates the term:

$$\bar{P}_e^{(\ell)} = \lim_{n\to\infty} E[P_e^{(\ell)}(g, \varepsilon, \alpha)].$$

The expectation is over the choice of code, channel noise realization, and missing wire realization.

Based on the proof of Thm. 2, we claim that the decoding error probability at any iteration ℓ for transiently and permanently missing connections, $\bar{P}_{eT}^{(\ell)}$ and $\bar{P}_{eP}^{(\ell)}$, becomes identical with the increase of the girth as blocklength n increases. In particular, in density evolution the state variable $x_{\ell+1}$ is computed based on the x_ℓ values of nodes immediately below in the infinite tree. Each connection in the

tree is encountered only once. In case of permanent failures each connection is present in the decoder with probability $1-\alpha$, whereas for transient failures each connection is present at any iteration with probability $1-\alpha$. But, for a given code symbol, its intrinsic messages traverse a particular edge only once if the LDPC graph is a tree. Thus the messages experience the same statistical effect under permanent and transient failures. This results in the same probability of error under both failures.

Theorem 4. *For any arbitrarily small $\delta = \delta(d_{\mathtt{v}}, d_{\mathtt{c}}, \ell) > 0$, $\sigma > 0$, and $\ell \geq 0$:*

$$\Pr[|\bar{P}_{eT}^{(\ell)} - \bar{P}_{eP}^{(\ell)}| \geq \sigma] \leq \delta.$$

Proof. First, let $N_{\vec{e}}^{2\ell}$ be the neighborhood of an edge $\vec{e}$ with depth 2ℓ in the decoding graph. Define the event A_N as $N_{\vec{e}}^{2\ell}$ is not tree-like. It is shown that for a positive constant $\tau = \tau(d_{\mathtt{v}}, d_{\mathtt{c}}, \ell)$, $\Pr[A_N] \leq \frac{\tau}{n}$ [31, Thm 2]. It implies the probability of exposing an edge multiple times decreases with increasing blocklength n at any iteration ℓ. Following the edge exposing procedure, $\bar{P}_{eT}^{(\ell)}$ and $\bar{P}_{eP}^{(\ell)}$ differ only when any edge $\vec{e}$ is exposed multiple times and the presence of $\vec{e}$ in the two decoding graphs with permanently and transiently missing connections differs. Hence, $\Pr\left[|\bar{P}_{eT}^{(\ell)} - \bar{P}_{eP}^{(\ell)}| \geq \sigma\right] = \Pr\left[|\bar{P}_{eT}^{(\ell)} - \bar{P}_{eP}^{(\ell)}| \geq \sigma \middle| A_N\right] \Pr[A_N] + \Pr\left[|\bar{P}_{eT}^{(\ell)} - \bar{P}_{eP}^{(\ell)}| \geq \sigma \middle| {A_N}^c\right] \Pr[{A_N}^c]$. Since $\Pr\left[|\bar{P}_{eT}^{(\ell)} - \bar{P}_{eP}^{(\ell)}| \geq \sigma \middle| {A_N}^c\right] = 0$, we can show $\Pr\left[|\bar{P}_{eT}^{(\ell)} - \bar{P}_{eP}^{(\ell)}| \geq \sigma\right] \leq 1 \cdot \Pr[A_N] \leq \frac{\tau}{n}$. As $n \to \infty$, this probability $\frac{\tau}{n} = \delta$ approaches 0. □

In the sequel, no distinction is made between the analyses for transiently and permanently missing connection cases.

2.4.5 Decoder Useful Region and Thresholds

Usually density evolution converges to a certain stable fixed point with increasing number of iterations ℓ. We define this fixed point as:

$$\bar{P}_e^{(\infty)} = \lim_{\ell \to \infty} \bar{P}_e^{(\ell)} = \lim_{\ell \to \infty} \lim_{n \to \infty} E[P_e^{(\ell)}(g, \varepsilon, \alpha)].$$

In order to decide when to use a decoder with missing connections, a *useful decoder* is defined. A decoder is said to be useful and should be used instead of

taking the codeword directly from the channel without decoding, if the asymptotic decoding error probability satisfies [1]:

$$\bar{P}_e^{(\infty)} < \bar{P}_e^{(0)} = \varepsilon.$$

The useful region of a decoder is defined as the set of parameters, in our case (ε, α), that satisfies the above condition. Note that in case of transient computation noise where computation is erroneous with probability $\tilde{\alpha}$ [1], there are $(\varepsilon, \tilde{\alpha})$ such that $\bar{P}_e^{(\infty)} > \varepsilon$. But under missing connections, for peeling, Gallager A, and Gallager B decoders, $\bar{P}_e^{(\infty)} \leq \varepsilon$ for any (ε, α). This is because these decoders do not propagate erroneous messages under missing connections and hence cannot degrade symbols received from the channel. When decoding with a fault-free decoder where $\alpha = 0$, there exists an ε^* below which the final decoding error probability goes to 0 and a much larger value otherwise. We will see in the following sections that $\bar{P}_e^{(\infty)}$ does not go to zero for positive α, but a threshold phenomenon still exists.[2] For every fixed α, there exists a channel noise decoding threshold ε^*, below which the final error probability $\bar{P}_e^{(\infty)}$ goes to a small value η. We call decoders that can achieve $\bar{P}_e^{(\infty)}$ that is lower than this small value η-reliable, and the channel noise level beyond which the decoder is not η-reliable the decoding threshold ε^* [1]:

$$\varepsilon^*(\eta, \alpha) = \sup\Big\{\varepsilon \in [0, 0.5] \Big| \bar{P}_e^{(\infty)} \text{ exists and } \bar{P}_e^{(\infty)} < \eta\Big\}.$$

[2]In general for $\bar{P}_e^{(\infty)}$ to go to 0 in a faulty decoder, one needs to either substantially change the decoder or to have a structural relationship between data and errors [17, 25, 28].

CHAPTER 3

PERFORMANCE OF DECODERS WITH MISSING CONNECTIONS

3.1 Peeling Decoder over Binary Erasure Channel

Consider the peeling decoder for communication over a BEC with alphabet $\{\pm 1, ?\}$. The check node computation is a product of all messages ± 1 it receives from neighboring variable nodes if none is "?", otherwise an erasure symbol "?" is sent. The variable node computation is to send any ± 1 symbol received either from the other check nodes or from the channel, otherwise send "?". When the connection between two nodes is missing, the message exchanged is equivalent to "?", so peeling extends naturally to decoders with missing connections. Note that this decoder satisfies the symmetry conditions C1–C4, so we can use density evolution assuming the all-one codeword was transmitted.

Although high-level intuition would suggest that the performance of decoding would degrade for any code and any decoder with missing connections, this is not the case as we later show for the Gallager B decoder. For the peeling decoder, the intuition holds and can be formalized using coupling techniques and the fact that peeling decoders never propagate erroneous messages.

Lemma 1. *For any LDPC code g with an arbitrary but finite blocklength, after a finite number of decoding iterations ℓ, for both permanently and transiently missing connections, the symbol error probability $P_e^{(\ell)}(g, \varepsilon, \alpha)$ increases monotonically with α for a given ε.*

Proof. Let us use P_e as shorthand for $P_e^{(\ell)}(g, \varepsilon, \alpha)$. The proof for monotonicity of P_e follows by simple coupling arguments. For a specific LDPC code, consider two different missing connection probabilities α_1 and α_2, where $\alpha_1 < \alpha_2$. Then, we couple the two missing connection processes as follows. Remove the wires with probability α_1, and from this check-variable connection graph, remove each of the remaining connections with probability $\alpha_2 - \alpha_1$. This gives a second missing connection process. It is not hard to check that the probability of missing connection

in the second process is α_2. Thus we can couple the missing connection processes to get a sample path dominance of connections. In this coupling, any realization of α_2 process has more missing connections than that of α_1.

Now consider the probability of correctly decoding any bit i. Note that with the peeling decoder, no erroneous messages are exchanged between the check and variable nodes; only correct messages and erasures are passed along wires. A variable node v_i holding message bit i can be decoded correctly if either the received bit is correct, or the received bit is an erasure but v_i receives a correct message through a path on the computation tree passing through one of its check nodes. The probability that the received bit is correct is the same in case of both α_1 and α_2. So, let us compare the other probability. Now, by coupling as any realization of α_1 has more connections than α_2, if a correct message reaches i following a path in the α_2 graph, then that path also exists in the α_1 graph. Thus, the event of receiving a correct message in case of α_2 is a subset of that of α_1. This proves the monotonicity of probability of correct decoding. Hence, missing connections can only degrade the performance. □

A similar coupling argument yields an ordering relationship with respect to channel erasure probability ε for a given α.

3.1.1 Density Evolution Equation

First, recall that the peeling decoding algorithm allows $\{\pm 1, ?\}$ to be sent, where "?" stands for an erasure caused by either the channel noise or a missing connection. In this case, the decoder only outputs either the correct message or an erasure symbol.

Consider a regular $(d_{\mathtt{v}}, d_{\mathtt{c}})$ LDPC code, BEC(ε), and each wire that can be disconnected independently with probability α. Let $x_0, x_1, \ldots, x_\ell$ denote the fraction of erasures existing in the code at each decoding iteration. The original received message from the channel is erased with probability ε, so

$$P_e^{(0)}(\varepsilon, \alpha) = x_0 = \varepsilon.$$

Let q_{in} be the probability that a node receives an erasure, and q_{out} be the probability that a node sends out an erasure. At a variable node, the probability that a given internal incident variable will be erased is the probability that both the ex-

ternal incident variable is erased and all other $d_{\mathsf{v}} - 1$ nodes are either disconnected or connected but erased.

$$\begin{aligned} q_{out} &= x_0 \sum_{i=0}^{d_{\mathsf{v}}-1} \binom{d_{\mathsf{v}}-1}{i} \alpha^i [q_{in}(1-\alpha)]^{(d_{\mathsf{v}}-1)-i} \\ &= \varepsilon[\alpha + (1-\alpha)q_{in}]^{d_{\mathsf{v}}-1}. \end{aligned}$$

At a check node, the probability that a given incident variable will not be erased is the probability that all $d_{\mathsf{c}} - 1$ other internal incident variables are not erased or disconnected. So the probability that a message is erased is

$$q_{out} = 1 - [(1-q_{in})(1-\alpha)]^{d_{\mathsf{c}}-1}.$$

Hence, the density evolution of the fraction of erasure between two consecutive decoding iterations is

$$x_{\ell+1} = \varepsilon\big[\alpha + (1-\alpha)\big(1 - [(1-x_\ell)(1-\alpha)]^{d_{\mathsf{c}}-1}\big)\big]^{d_{\mathsf{v}}-1}.$$

The density evolution result can be extended to irregular LDPC codes:

$$x_{\ell+1} = \varepsilon\lambda\Big(\alpha + (1-\alpha)\big(1 - \rho[(1-x_\ell)(1-\alpha)]\big)\Big).$$

Let $f_{DE}(x_\ell, \varepsilon, \alpha) = x_{\ell+1}$ be the recursive update function for the fraction of erasure, where $0 \leq \varepsilon < 0.5$ and $0 \leq \alpha \leq 1$ is the domain of interest.

3.1.2 Fixed Points

The density evolution function f_{DE} is non-decreasing in each of its arguments, given the other two. Thus, a monotonicity result similar to [32, Lem. 3.54] holds here. This also implies that a convergence result for x_ℓ, similar to [32, Lem. 3.56], holds. So, for a given α and ε, x_ℓ converges to the nearest fixed point of $x = f_{DE}(\varepsilon, x, \alpha)$. Due to this existence of the fixed point, we can characterize the error probability when the decoding process is finished. The fixed points can be found by solving for the real solutions to the polynomial equation

$$x - \varepsilon\lambda\Big(\alpha + (1-\alpha)\big(1 - \rho[(1-x)(1-\alpha)]\big)\Big) = 0. \tag{3.1}$$

We now prove that the decoding error probability is strictly positive by showing that $x = 0$ is not a fixed point in (3.1).

Lemma 2. *For any irregular code ensemble $C^{\infty}(\lambda, \rho)$, there exists a $\delta > 0$, such that the probability of error $P_e^{(\infty)}$ satisfies $P_e^{(\infty)} > \varepsilon\lambda(1 - (1 - \alpha)\rho(1 - \alpha)) > \delta > 0$.*

Proof. Since x_ℓ is monotonic, if $x_0 \leq x_1$ then for any ℓ, $x_{\ell+1} \geq x_\ell \geq x_{\ell-1}$. Now, for $x_0 = 0$, by substituting this value in f_{DE},

$$x_1 = f_{DE}(0, \varepsilon, \alpha) = \varepsilon\lambda(1 - (1 - \alpha)\rho(1 - \alpha)) > 0 = x_0.$$

This implies that $\lim_{\ell\to\infty} x_\ell \geq f_{DE}(0, \varepsilon, \alpha)$, for $x_0 = 0$. But, as x_ℓ converges to the fixed point nearest to x_0 in the direction of monotonicity, $x = 0$ is not a fixed point and there is no fixed point in $(0, f_{DE}(0, \varepsilon, \alpha))$ for any $\varepsilon, \alpha > 0$. Thus we have $P_e^{(\infty)} > 0$. □

Since this lemma shows all fixed points of the density evolution equation are greater than $\varepsilon\lambda(1 - (1 - \alpha)\rho(1 - \alpha))$, decoding error probability cannot be taken to zero. But this does not mean that the decoder is not useful. In fact it is always better to use the decoder, even when there are missing connections, rather than just taking corrupted symbols from the channel directly, since the peeling decoder never has incorrect messages. We can see this using the monotonicity of $f_{DE}(x, \varepsilon, \alpha)$ in each of its arguments, given the other two. For any channel and code, $x_0 = \varepsilon$, and it follows from f_{DE} that $x_1 = f_{DE}(x_0, \varepsilon.\alpha) \leq \varepsilon$. Hence $x_\ell \leq \varepsilon$, for all ℓ, and $P_e^{(\infty)} \leq \varepsilon$. This is in sharp contrast to the decoders with computation noise, where decoder output can be strictly worse than channel output [1].

3.1.3 Performance Analysis

In the previous section, we developed the recursive function to characterize the final error probability achieved by a peeling decoder with missing wires. Now we want to characterize the performance of such decoders.

For a peeling decoder, when $\varepsilon = 0$, the error probability stays at 0 regardless of the quality of the decoder. When $\alpha = 0$, it has been shown that there exists decoding threshold on the channel noise ε, below which the final error probability

can be driven to 0 with the increase of decoding iterations [31]. For the following analysis, we consider the system when $\varepsilon > 0$ and $\alpha > 0$. Ideally, we want the error probability to be driven to 0, but as demonstrated in Lem. 2, this is impossible. Here we use the weaker notion of η-reliability defined in Sec. 2.4.5, where η limits the final decoding error probability P_e.

Figure 3.1 shows the final symbol error rate of decoding a $C^{\infty}(3, 6)$ LDPC code under peeling decoding with various missing connection probabilities α over BEC(ε). It can be seen that given α, there exists a threshold in channel noise level where a phase transition in P_e happens. Figure 3.2 illustrates such thresholds with the change of α under different small η-reliable constraints. An interesting phenomenon to notice in the decoding threshold is that there also exists a phase transition with the change of the decoder missing connection probability α. With the increase of α, for a fixed η-reliable decoder with missing connections, the decoding threshold first decreases linearly, and then exhibits more rapid decrease before convergence to zero.

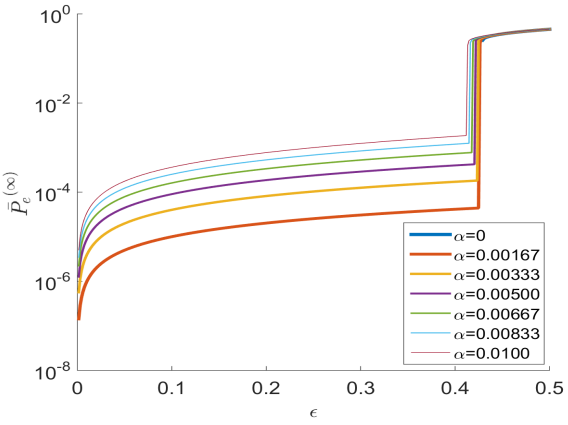


Figure 3.1: Final symbol error rate of decoding a $C^{\infty}(3, 6)$ LDPC code under peeling decoding algorithm with various missing connection probability α over BEC.

3.2 Gallager A Decoder over Binary Symmetric Channel

Consider a fault-free Gallager A decoder for communication over a BSC. The messages are passed along the edges in the corresponding Tanner graph during decoding. A check node computes the product of incoming variable-to-check

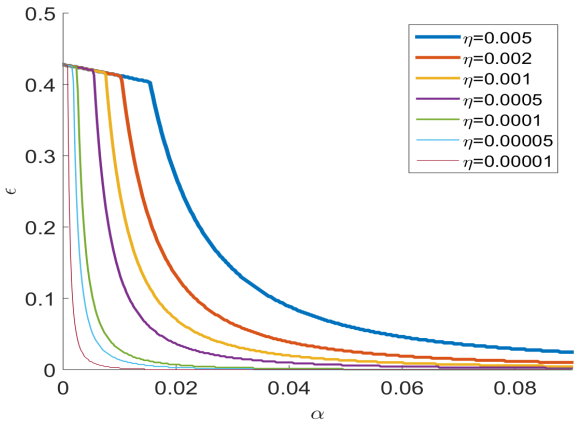


Figure 3.2: Channel threshold of decoding a $C^{\infty}(3,6)$ LDPC code under peeling decoding algorithm over BEC(ε) for different given final error η-limits.

node messages $\{u_{v\to c}\}$; a variable node decides to flip the message from channel y_v if all of the incoming check-to-variable node messages are $-y_v$ [3].

With the introduction of missing connections, the check node computation is not defined if an input is unknown ("?"). The product computed at the check node is the modulo-2 sum of all incoming messages to ensure that the parity constraints of the code are satisfied. When one of the bits involved in the parity is unknown, that parity check is no longer informative. This is because any bit of a linear code is equally likely to be ± 1 (as complementing a binary codeword gives a codeword). So, for decoders with missing connections we make a natural adaptation: $u_{c\to v} =$ "?" if any of the incoming messages- is "?". We also make a natural adaptation for variable node computation: $-y_v$ is sent if more than one non-erasure check node messages are $-y_v$, and y_v is sent otherwise.

When it comes to Gallager A decoding over BSC, the messages being passed between nodes may carry erroneous information, unlike the peeling decoder for BEC, where the messages are either correct or erasure. So, for a sample path realization of channel and missing connections, a missing connection may prevent propagation of erroneous messages. Hence, unlike the peeling decoder for BEC, it is not apparent that there exists a stochastic dominance result like Lem. 1 between two different probabilities of missing connections. As fault-free decoding with the Gallager A algorithm satisfies conditions C1–C4, we can restrict analysis to the all-one codeword.

3.2.1 Density Evolution Equation

We find the probability for a variable node to compute -1 at iteration $\ell + 1$, in terms of x_ℓ. We consider a regular $(d_\mathsf{v}, d_\mathsf{c})$ LDPC code and the adaptation of Gallager A decoding with erasure symbols for missing connections.

First note that since a BSC only outputs ± 1, a variable node never computes "?" with the Gallager A adaptation, even though it may receive (due to connection failure or check-node computes "?") or send the erasure symbol "?" (only due to connection failure).

The probability that a check node computation is -1 is:

$$\begin{aligned}
&\Pr\left\{\text{all } (d_\mathsf{c}-1) \text{ variable nodes are connected and send odd number of } -1\right\} \\
&= (1-\alpha)^{d_\mathsf{c}-1} \Pr\{\text{odd number of } (d_\mathsf{c}-1) \text{ nodes send } -1\} \\
&= (1-\alpha)^{d_\mathsf{c}-1} \frac{(1-(1-2x_\ell)^{d_\mathsf{c}-1})}{2},
\end{aligned}$$

where the last line follows using results from [3, Sec. 4.3].

The probability that a check node computation is $+1$ is:

$$\begin{aligned}
&\Pr\left\{\text{all } (d_\mathsf{c}-1) \text{ variable nodes are connected and send even number of } -1\right\} \\
&= (1-\alpha)^{d_\mathsf{c}-1} \Pr\{\text{even number of } (d_\mathsf{c}-1) \text{ nodes have } -1\} \\
&= (1-\alpha)^{d_\mathsf{c}-1} \frac{(1+(1-2x_\ell)^{d_\mathsf{c}-1})}{2}.
\end{aligned}$$

The probability that a check-to-variable message is "?" is the complement of the probability that a check node computes ± 1. Define p_0 to be

$$1-(1-\alpha)^{d_\mathsf{c}-1}. \tag{3.2}$$

Consider a random variable $V \sim \text{Binomial}(d_\mathsf{v}-1, 1-\alpha)$ with probability mass function $p_V(v)$, capturing the distribution of number of check nodes connected to a variable node. Define p_{+1} and p_{-1} such that

$$p_{+1} = (1-\alpha)^{d_\mathsf{c}-1} \frac{(1+(1-2x_\ell)^{d_\mathsf{c}-1})}{2} \tag{3.3}$$

and

$$p_{-1} = (1-\alpha)^{d_\mathsf{c}-1} \frac{(1-(1-2x_\ell)^{d_\mathsf{c}-1})}{2}. \tag{3.4}$$

Now consider $x_{\ell+1}$, the error probability at a variable node at the $(\ell+1)$th iteration. The fraction of incorrect values held at this variable node is the sum of the probability of two events. The first event is that the message received from the channel is correct, and none of the incoming messages from the connected check nodes is correct, but not all of them are "?", and not only one says different while others are "?". The second event is that the message received from the channel is wrong, and at least one of the incoming messages from the connected check nodes is wrong or at most one check node is correct while all others are "?".

The probability of the first event is:

$$\begin{aligned}
&\mathbb{E}_V \Big[(1-\varepsilon)\big[\Pr\{\text{no connected check nodes sends } 1\} \\
&\quad - \Pr\{\text{all } V \text{ connected check nodes send "?"}\} \\
&\quad - \Pr\{\text{one check node sends } -1 \text{ while others send "?"}\}\big]\Big] \\
&= \sum_{v=1}^{d_v-1} p_V(v)(1-\varepsilon)[(p_{-1}+p_0)^v - p_0^v - p_{-1}{p_0}^{v-1}].
\end{aligned}$$

The probability of the second event is:

$$\begin{aligned}
&\mathbb{E}_V \Big[\varepsilon\big[\Pr\{\text{at least one connected check nodes send } -1\} \\
&\quad + \Pr\{\text{all } V \text{ connected check nodes send "?"}\} \\
&\quad + \Pr\{\text{one check node sends } +1 \text{ while others send "?"}\}\big]\Big] \\
&= \mathbb{E}_V \Big[\varepsilon\big[1 - \Pr\{\text{no connected check nodes sends } -1\} \\
&\quad + \Pr\{\text{all } V \text{ connected check nodes send "?"}\} \\
&\quad + \Pr\{\text{one check node sends } +1 \text{ while others send "?"}\}\big]\Big] \\
&= \sum_{v=0}^{d_v-1} p_V(v)\varepsilon[1-(p_{+1}+p_0)^v + p_0^v + p_{+1}{p_0}^{v-1}].
\end{aligned}$$

Let $x_{\ell+1} = f_{DE}(x_\ell, \varepsilon, \alpha)$, and take the expectation of V according to the bino-

mial distribution to get

$$\begin{aligned} x_{\ell+1} &= f_{DE}(x_\ell, \varepsilon, \alpha) \\ &= \varepsilon\alpha^{d_{\mathrm{v}}-1} + \sum_{v=1}^{d_{\mathrm{v}}-1} \binom{d_{\mathrm{v}}-1}{v} (1-\alpha)^v \alpha^{(d_{\mathrm{v}}-1-v)} \Big[(1-\varepsilon)[(p_{-1}+p_0)^v - p_0^v \\ &\quad - p_{-1}{p_0}^{v-1}] + \varepsilon[1-(p_{+1}+p_0)^v + p_0^v + p_{+1}{p_0}^{v-1}] \Big]. \end{aligned}$$

To extend to irregular LDPC ensembles, we take the average of the check node distribution and get:

$$p_{+1}^{(irr)} = \rho(1-\alpha)\frac{1-\rho(1-2x_\ell)}{2} \tag{3.5}$$

and

$$p_{-1}^{(irr)} = \rho(1-\alpha)\frac{1+\rho(1-2x_\ell)}{2}. \tag{3.6}$$

The terms in $f_{DE}(x_\ell, \varepsilon, \alpha)$ have to be averaged over the variable node degree distribution of d_{v} with function $\lambda(\cdot)$.

3.2.2 Fixed Points

It can be seen that $f_{DE}(x, \varepsilon, \alpha)$ is monotonic in x for a set of given α and ε. Hence, by the same arguments as for peeling decoders, for any initial $0 \leq \varepsilon = x_0 \leq 0.5$, x_ℓ converges to the nearest fixed point of the density evolution equation. We use $\tau_1 \leq \tau_2 \leq \tau_3 \leq \cdots$ to denote these fixed points.

Note that for all $\varepsilon > 0, \alpha > 0$, and $x_\ell = 0$, $f_{DE}(x_\ell, \varepsilon, \alpha) = x_{\ell+1} > 0$. This implies a result similar to Lem. 2 here. With the existence of channel noise and missing wiring, the decoding probability cannot be driven to 0. It is easy to show that for $\varepsilon = 0$, $f_{DE}(x, 0, \alpha)$ has one fixed point at $\tau_1 = 0$. We then focus on the case where $0 < \varepsilon < 0.5$ for the following analysis.

Define $p^+(x) = (p_{-1}+p_0)^v - p_0^v - p_{-1}{p_0}^{v-1}$ and $p^-(x) = 1-(p_{+1}+p_0)^v + p_0^v$. An analytical expression for the channel threshold is the root (τ_2) of the following expression between 0 and 0.5:

$$x\lambda(\alpha) + \lambda\Big(p^+(x) - xp^+(x) + xp^-(x)\Big) = x.$$

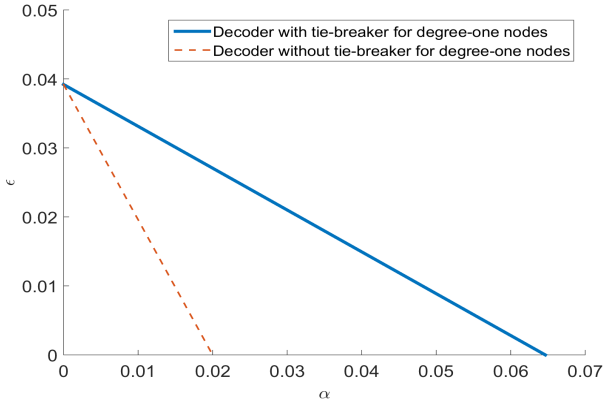


Figure 3.3: Decoding a $C^{\infty}(3,6)$ regular LDPC code with α-missing wire Gallager A decoding algorithm over BSC(ε). The useful region where it is beneficial to use decoder is between the curve and α-axis.

The solid line in Fig. 3.3 shows the useful region of decoding for a $(3,6)$ regular LDPC code with missing wire, which is between τ_1 and τ_2 due to the monotonicity of function f_{DE}. Compared to [1, Fig. 2] where computation at each node is noisy with probability α, the useful region of a decoder with missing connection is larger. In this case, decoders with missing connections outperform those with noisy internal computation. At any node, if the corresponding incoming message is missing rather than noisy with probability α, the node is more likely to send a correct message than an erroneous one.

3.2.3 Performance Analysis

Figure 3.4 shows η-thresholds for communication over BSC(ε) with a Gallager A decoder with missing connections. Recall that for a $(3,6)$ regular LDPC code with a fault-free Gallager A decoder, the threshold is roughly 0.039 [34]. Note that P_e can be driven to a fairly small number even with missing wires. Decoding is robust to missing connection defects, though less than the peeling decoder over BEC.

As observed in Fig. 3.4, a phase transition of the decoding threshold ε with the change of missing connection probability α noticed in the peeling decoder case also exists here.

In contrast to classic settings, there may be degree-one nodes in decoding graphs due to the random missing connections. Hence, a tie-breaker at a variable node is

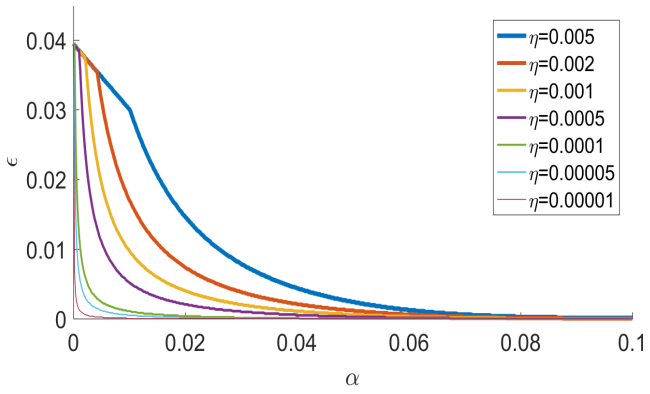


Figure 3.4: Channel threshold of decoding a $C^{\infty}(3,6)$ LDPC code under Gallager A decoding algorithm over BSC(ε) for different given final error η-limits.

necessary when the only incoming message from a check node is different from the received message from the channel. Since the channel message is more reliable than internal messages when there are missing connections in the decoder, we choose not to flip the channel message when the only incoming non-erasure message is the opposite. With this minor twist, the decoding threshold increases significantly. The dotted line in Fig. 3.3 shows the useful region of the decoder without the tie-breaker for degree-one case, choosing to flip the channel symbol when all incoming non-erasure messages are different from the channel symbol.

3.3 Gallager B Decoder over Binary Symmetric Channel

Gallager B decoders are usually more robust than Gallager A decoders without missing connections [18], so we modify the Gallager B algorithm by introducing erasure symbols for missing connections. In the Gallager B decoder, a check node performs the same operation with incoming variable-to-check node messages as Gallager A in Sec. 3.2, sending an unknown symbol "?" if one of the incoming messages is from a disconnected node. At a variable node however, instead of flipping the current value u only when all the incoming messages from connected nodes say $-u$, a variable node in the Gallager B decoder decides to correct the current value u when there are more than b number of incoming messages that are $-u$. This threshold can be iteration-specific to reach optimality. In our case, we use the majority criterion and choose $b^* = \lfloor \frac{d_v+1}{2} \rfloor$ in all iterations because

this threshold results in small error probability independent of iteration number in fault-free Gallager B work [35, Sec. 5]. We also choose b^* based on the designed code without counting the number of actually connected nodes for simplicity, and it is verified numerically that there is no significant difference in performance.

Similar to the Gallager A model developed in Sec. 3.2, the codeword symmetry conditions C1–C4 are all satisfied in invoking Prop. 1.

3.3.1 Density Evolution Equation

The density evolution equation for the Gallager B decoder is similar to Gallager A. Consider a regular $(d_{\mathsf{v}}, d_{\mathsf{c}})$ LDPC code and all-one codeword transmitted over BSC. At iteration ℓ, the probability of a check-to-variable message is "?", $+1$ or -1 with probabilities p_0, p_{+1} and p_{-1}, respectively, which have the same expressions as in Sec. 3.2.

Now consider $x_{\ell+1}$, the error probability at a variable node at the $(\ell+1)$th iteration. The fraction of incorrect values held at this variable node is the sum of the probability of two events. The first event is that the message received from the channel is correct, and at least $b = \lfloor \frac{d_{\mathsf{v}}+1}{2} \rfloor$ check nodes are connected and send incorrect messages. The second event is that the message received from the channel is wrong, and at most $b-1 = \lfloor \frac{d_{\mathsf{v}}-1}{2} \rfloor$ of the incoming messages from the check nodes are correct. Consider a random variable $V \sim \text{Binomial}(d_{\mathsf{v}}-1, 1-\alpha)$ capturing the distribution of the number of check nodes connected to a variable node.

The probability of the first event is:

$$\begin{aligned}
&\mathbb{E}_V \left[(1-\varepsilon) \Pr\{\text{at least } b \text{ check nodes are connected and send} - 1\} \right] \\
&= \sum_{v=b}^{d_{\mathsf{v}}-1} p_V(v) (1-\varepsilon) {p_{-1}}^v (1-p_{-1})^{d_{\mathsf{v}}-1-v}.
\end{aligned}$$

The probability of the second event is:

$$\begin{aligned}
&\mathbb{E}_V \left[\varepsilon \Pr\{\text{at most } (b-1) \text{ check nodes send} + 1\} \right] \\
&= \mathbb{E}_V \left[\varepsilon [1 - \Pr\{\text{at least } b \text{ check nodes are connected and send} + 1\}] \right] \\
&= \sum_{v=b}^{d_{\mathsf{v}}-1} p_V(v) \varepsilon [1 - {p_{+1}}^v (1-p_{+1})^{d_{\mathsf{v}}-1-v}].
\end{aligned}$$

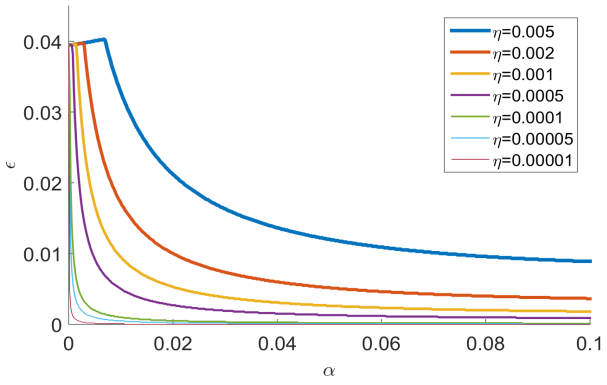


Figure 3.5: Channel threshold of decoding a $C^{\infty}(3,6)$ LDPC code under Gallager A decoding algorithm over BSC(ε) for different given final error η-limits.

Taking the expectation of V according to the binomial distribution, we have

$$x_{\ell+1} = \sum_{v=b}^{d_{\mathrm{v}}-1} \binom{d_{\mathrm{v}}-1}{v} (1-\alpha)^{v} \alpha^{(d_{\mathrm{v}}-1-v)} \Big[(1-\varepsilon)[{p_{-1}}^{v}(1-p_{-1})^{d_{\mathrm{v}}-1-v}] + \varepsilon[1-{p_{+1}}^{v}(1-p_{+1})^{d_{\mathrm{v}}-1-v}] \Big].$$

The density evolution equation can also be extended to irregular LDPC codes, with changes in parameters $b(x) = \lfloor \frac{\lambda(x)+1}{2} \rfloor$, $p_{+1}^{(irr)}$, and $p_{-1}^{(irr)}$ defined in expressions (3.5) and (3.6).

3.3.2 Performance Analysis

We carry out detailed performance characterization of the Gallager B decoder with missing connection and show that such a decoder is indeed more robust to missing connections than Gallager A.

Note that when variable node degree $d_{\mathrm{v}} = 3$ for a regular LDPC code, a fault-free Gallager B decoder with the defined threshold $b = \lfloor \frac{d_{\mathrm{v}}+1}{2} \rfloor$ is equivalent to a fault-free Gallager A decoder. However, due to the modification of the Gallager A decoder to keep the received channel message when there exists only one incoming message, in the case of missing connections, these two decoders behave differently for decoding a $C^{\infty}(3,6)$ regular LDPC code.

One interesting phenomenon shown in Fig. 3.5 is that the decoding thresholds first increase with the increasing decoder missing connection probability. This

error enhancement phenomenon is introduced by the missing connections, essentially resulting in a change of choice for threshold b in each iteration to achieve a lower error rate. This SF phenomenon demonstrates that optimization of degree distribution and threshold b in each iteration can be utilized to combat missing connections. A similar SF result shows that the errors introduced in estimating Markov random field models can be partially canceled and benefit end-to-end inference performance [36]. SF effects due to noise in computational elements, rather than graphical model structure errors as here, have been observed in [27, 29, 37] and later specifically in LDPC decoders [23, 25].

CHAPTER 4

MOVING TOWARDS PRACTICE

Though performance analysis of LDPC decoders with missing connections using density evolution is an important topic in coding theory, our eventual goal is to use analytical understanding for practical system design.

Towards this, we first briefly discuss how one can use DE analysis to study sensitivity of codes and decoders, so as to give insight into resource allocation over the entire telecommunications system. In particular we ask whether more resources should be spent in manufacturing or in operation. Second, as density evolution analysis is an asymptotic approximation of practical finite-length codes, we also perform simulations to understand how well the asymptotics describe finite-length code performance. Finally we note that increasing the accuracy of semiconductor fabrication by just a small amount requires a significant increase in manufacturing cost (which already takes tens of billions of dollars to build facilities, and limits growth of the industry). As such, we perform some preliminary manufacturing yield analysis to show the industry potential for our research. For brevity, this section is largely restricted to Gallager A.

4.1 Decoder Sensitivity Analysis

Should the industry invest more resources in operating good communication channels or in manufacturing better receiver hardware?

By taking the derivatives of the density evolution function $x_{\ell+1} = f_{DE}(x_l, \varepsilon, \alpha)$ with respect to ε and α and evaluating at $\bar{P_e}^{(\infty)} = x_\ell = x_{\ell+1}$, we can describe the impact of noise level of the channel and the missing connection on the final error rate.

$$\bar{P_e}^{(\infty)}(\varepsilon, \alpha) = \varepsilon\lambda(\alpha) + \lambda\left((1-\varepsilon)p^{+}(\bar{P_e}^{(\infty)}(\varepsilon, \alpha)) + \varepsilon p^{-}(\bar{P_e}^{(\infty)}(\varepsilon, \alpha))\right).$$

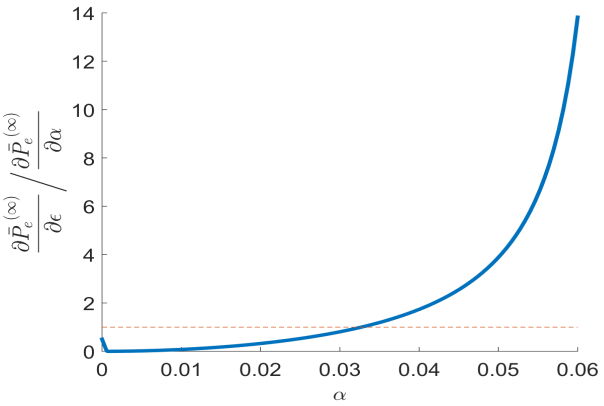


Figure 4.1: Comparison between the derivative of $\bar{P}_e{}^{(\infty)}(\varepsilon, \alpha)$ with respect to ε and α of decoding a $C^{\infty}(3, 6)$ regular LDPC code with α-missing wire Gallager A decoding algorithm over BSC(ε), when ε and α are at the boundary of decoder useful region.

Denote

$$g(x) = (1-\varepsilon)p^{+}(x(\varepsilon,\alpha)) + \varepsilon p^{-}(x(\varepsilon,\alpha)).$$

Take partial derivatives of each side with respect to ε:

$$\begin{aligned}\frac{\partial x(\varepsilon,\alpha)}{\partial \varepsilon} &= \lambda(\alpha) + \frac{\partial \lambda(g(x(\varepsilon,\alpha)))}{\partial x(\varepsilon,\alpha)}\frac{\partial x(\varepsilon,\alpha)}{\partial \varepsilon} + \frac{\partial \lambda(g(x(\varepsilon,\alpha)))}{\partial \varepsilon} \\ &= \frac{\lambda(\alpha) + \frac{\partial \lambda(g(x(\varepsilon,\alpha)))}{\partial \varepsilon}}{1 - \frac{\partial \lambda(g(x(\varepsilon,\alpha)))}{\partial x(\varepsilon,\alpha)}}.\end{aligned}$$

Similarly,

$$\begin{aligned}\frac{\partial x(\varepsilon,\alpha)}{\partial \alpha} &= \varepsilon\frac{\partial \lambda(\alpha)}{\partial \alpha} + \frac{\partial \lambda(g(x(\varepsilon,\alpha)))}{\partial x(\varepsilon,\alpha)}\frac{\partial x(\varepsilon,\alpha)}{\partial \alpha} + \frac{\partial \lambda(g(x(\varepsilon,\alpha)))}{\partial \alpha} \\ &= \frac{\varepsilon\frac{\partial \lambda(\alpha)}{\partial \alpha} + \frac{\partial \lambda(g(x(\varepsilon,\alpha)))}{\partial \alpha}}{1 - \frac{\partial \lambda(g(x(\varepsilon,\alpha)))}{\partial x(\varepsilon,\alpha)}}.\end{aligned}$$

Figure 4.1 illustrates the ratio of the derivative of $\bar{P}_e{}^{(\infty)}(\varepsilon, \alpha)$ with respect to ε and α, when α and ε are at the boundary of useful region depicted in Fig. 3.4:

$$\left(\frac{\partial \bar{P}_e{}^{(\infty)}(\varepsilon,\alpha)}{\partial \varepsilon} \Big/ \frac{\partial \bar{P}_e{}^{(\infty)}(\varepsilon,\alpha)}{\partial \alpha}\right).$$

Contrary to our intuition, both derivate values are negative at the boundary of the

useful region. Recall the linear relationship of ε and α at the boundary of the useful region; with the increase of α, ε has to decrease in order to stay in the useful region, resulting in the decrease in $\bar{P}_e^{(\infty)}(\varepsilon, \alpha)$.

When operating at the edge of the useful region, as we can see in Fig. 4.1, it is advantageous to put resources into circuit manufacturing up to an α value of roughly 0.03 where the curve crosses the equal-ratio point, whereas it is advantageous to put resources into the channel thereafter. Thus aiming for manufacturing that achieves such a crossover point α may be an appropriate resource allocation strategy.

4.2 Finite-length Simulations

As shown in [1, 31] and Sec. 2.4 in this thesis, the density evolution technique is based on the concentration around the cycle-free case when the code's blocklength n goes to infinity. However, codes of finite-length are not cycle-free and have positive probability of edge repeats in the unrolled computation tree. In order to demonstrate the significance of the results derived with density evolution, we simulate finite-length codes for noisy decoders described in [1] and decoders with missing connections and compare the two types of faulty decoders.

4.2.1 Noisy Decoder for Finite-length Codes

As discussed in Ch. 2, one type of defective decoder is that with noisy computation elements, where the connection diagram of the wires is drawn correctly. At each node in a message-passing decoder, a message is sent out with processing error independently from other nodes and iterations. The performance of such noisy LDPC decoders is characterized in [1] using density evolution. Here, we simulate the finite-length systems with transiently noisy Gallager A decoders to demonstrate that performance is close to the analytical results derived with density evolution.

For $(3, 6)$-regular LDPC codes with blocklength $n = 198$, 498, 1998, and 4998 drawn at random from the code ensemble using socket-switching, the decoding performance is simulated with transiently noisy decoders that randomly flip each correctly computed message with probability $\alpha = 0.005$ at every iteration. For each trial, decoding is performed for 25 iterations (error probability usually con-

vergences within 10 iterations in fault-free decoding). For each channel noise level and decoder noise level, the decoding error probability is averaged over 300 randomly selected code realizations.

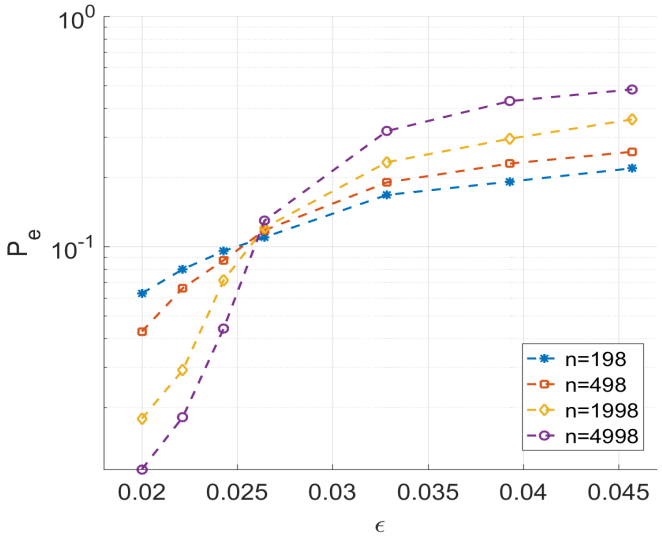


Figure 4.2: P_e of decoding $(3, 6)$ LDPC code with finite blocklength under Gallager A decoding algorithm over BSC with $\alpha = 0.005$ noisy decoder.

As Fig. 4.2 illustrates, the symbol error rates for all blocklengths increase smoothly with channel noise. According to the analytical results in [1], the decoding threshold ε^* for $(3, 6)$-regular LDPC code with $\alpha = 0.005$ noisy decoder is roughly 0.026, which is where P_e for different blocklengths intersect in Fig. 4.2. As observed, the symbol error rate P_e decreases as blocklength n increases when channel noise is below the decoding threshold, while P_e exhibits the opposite trend when the channel noise is above the decoding threshold. The transition around the decoding threshold also becomes sharper as the blocklength grows larger, similar to the trend in fault-free decoding simulation results in [38], indicating that the performance of finite-length codes approaches the density evolution analysis as the blocklength n goes to infinity.

4.2.2 Decoder with Missing Connections for Finite-Length Codes

We simulate finite-length systems having decoders with either transiently or permanently missing connections, to demonstrate performance is comparable in the two settings and predicted by density evolution. For $(3, 6)$-regular LDPC codes

with blocklength $n = 498$, 1002, and 1998 drawn at random from the code ensemble using socket-switching, we randomly simulate decoding performance with connections either permanently removed before the decoding starts or transiently removed during each decoding iteration for various sets of (ε, α). For each trial, decoding is performed for more than 30 iterations. For each channel noise level and missing connection probability, the decoding error probability is averaged over 100 randomly selected code realizations and missing connection realizations.

As Fig. 4.3 illustrates, the performance of finite-length codes resembles the asymptotic performance of codes. As expected, below the decoding threshold, the symbol error rates of finite-length codes are higher than asymptotic performance. For fault-free decoders with channel noise below threshold, the asymptotic symbol error rate is 0, whereas in the case of finite-length codes P_e increases smoothly with increasing ε [38]. Similar to fault-free decoders for finite-length codes, decoders with missing connections show a similar trend of increasing P_e. Further, as observed in the simulations, the performances of transiently and permanently missing connection cases are close to one another. We chose $\alpha = 0.02$ because it is within the range of defective interconnect 1–15% estimated in [8]; see also Sec. 4.1. Different from [38], in the simulations, codes with small stopping sets are not expurgated. Also recall from Thm. 1 that the concentration of the individual performance around the ensemble average is exponential in blocklength and the concentration happens more slowly in the case of missing connections compared to fault-free decoders. Hence, there is more numerical variation in the simulation results at all blocklengths, especially for small n. Nevertheless, simulations show that the asymptotic analysis of decoders with missing connections has practical significance.

It is also observed that when the channel is noisy (around and above decoding threshold), decoders with missing connections outperform noisy decoders for finite-length codes under Gallager A algorithm even when the missing connection probability 0.02 is much higher than the decoder noise 0.005. This further shows that under some conditions, passing missing connections or no information is better than noisy or wrong information.

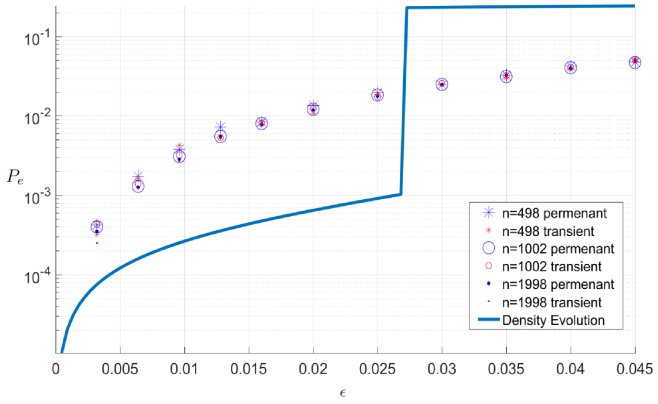


Figure 4.3: P_e of decoding $(3, 6)$ LDPC code with finite blocklength under Gallager A decoding algorithm over BSC with permanently and transiently missing connection probability $\alpha = 0.02$.

4.3 Semiconductor Manufacturing Yield Analysis

By understanding the computational purpose of circuits (here decoding) it is often possible to raise effective manufacturing yield above the raw yield where all components must be fault-free [39, 40]. To demonstrate that the effective yield of LDPC decoder circuits increases by allowing missing connection probability α that still guarantees the decoding performance η, we can apply the error-tolerant methodology [39,40]. Threshold testing in [39] accepts or rejects a chip based on whether the chip's performance passes a determined threshold; the increase in effective yield is the amount of chips with defects but still meeting the performance requirement. In our case of decoders with missing connections, this threshold is the maximum symbol error rate η. We want to find the highest missing connection probability $\alpha_{max}(G^n, \varepsilon, \eta)$ such that for every decoder with $\alpha \leq \alpha_{max}$, the resulting P_e is under the target error rate η for a given code ensemble G^n and channel quality ε.

Let $\phi(\alpha)$ be the yield factor, the expected percentage of decoders with missing connection probability α, and $p(\alpha)$ be the probability that the circuit has defect density α, often taken as an exponential distribution [40]. Then the effective yield is:

$$Y = \int_0^{\alpha_{max}(G^n,\varepsilon,\eta)} p(\alpha)\phi(\alpha)d\alpha.$$

For the $\mathcal{C}^\infty(3, 6)$ LDPC code ensemble and $\eta = 10^{-5}$, for a large range of possible channel values ε, Fig. 3.2 shows us that $\alpha_{max} = 0.01$ is more than sufficient for

the case of peeling decoder under BEC. It is straightforward to see that, compared to the yield of the fault-free case $Y_0 = p(0)\phi(0)$, allowing some error-tolerance in manufacturing may increase effective yield significantly. For the exponential distribution function, the absolute increase in yield is linear in α_{max} [40]:

$$\Delta Y = Y - Y_0 = \alpha_{max} \frac{D_0 A}{(1 + AD_0)^2},$$

where D_0 is the defect density (average number of defects per unit of chip area), and A is the chip area. Likewise the fractional increase in yield, is:

$$\Delta Y / Y_0 = \frac{Y - Y_0}{Y_0} = \alpha_{max} \frac{D_0 A}{1 + AD_0}.$$

As shown in the previous sections, a small defect rate α does not degrade the performance too much. However, as reported in the semiconductor manufacturing industry, a 1% reduction in yield can result in a 12% reduction in profit [41, 42]. Hence even by allowing a small probability of defects α, the industry can save a significant amount of wastage and cost without much change in performance.

CHAPTER 5

COMPLEX WEIGHTED NETWORKS

Networked structure, a graphical model with a set of nodes and edges, is a complex and useful topological model where weights are assigned to the links in the connection graph to accurately capture the intensity of interactions in both natural and man-made systems [43–49]. With the growing popularity of data mining, social networks, and biological networks, stochastic diffusion models based on Markov chains have been the focus of a great deal of attention due to their applications to analyze the seemingly chaotic data in various fields. Pons and Latapy [50] computed communities in large-scale networks using random walks; Blanchard and Volchenkov [51] gave an extensive overview on the mathematics behind random walks and diffusion on networks, using examples that range from the study of epidemics, to synchronization and self-regulation in complex networks.

Semi-metrics, distance functions that violate the triangle inequality, only exist in networks that lie in non-metric spaces. Consider a network with a pair of elements x_i and x_j: it might happen that the distance of the direct path between these two elements, d_{ij}, is larger than the distance of an indirect path, $d_{ik} + d_{kj}$, through a third element x_k. The intensity of semi-metric behavior is computed by the ratio of direct path distance over indirect path distance. Hence, this measure of semi-metricity captures the latent association among the elements in a network.

Most of the works on networks fit the data collected to a stochastic Markov chain in a metric space by approximately calculating the transition probability from one state to another with the normalizing constraint $\sum_j p_{ij} = 1$, namely, that the sum of the probabilities from state i to all possible states j must be 1. The equilibrium distribution, where the changes in the network quiet down, can then be derived by finding the convergence of the transitions. While this method has assisted researchers immensely in deriving meaningful results, the semi-metric property is lost in such models, since the semi-metric property only holds in non-metric distance spaces. With a stochastic model, the probability distribution of all states at every unit time is constrained to $\sum_j p_{ij} = 1$, where the triangle inequality

holds everywhere.

We propose an algebraic method to calculate the equilibrium distribution of a stochastic diffusion model via the transitive closure based on the interaction graph obtained directly from the observed data without the normalizing constraint in the stochastic model. With this proposed algebraic method, the semi-metric property can be preserved. Moreover, since semi-metric edges can be treated as redundant connections in a network, removing these semi-metric edges in the network can shorten the computation for various algorithms. Learning more about the algebraic representation behind the n-diffusion model would also provide us with more insights into the phenomenon. In the following sections, we establish the connection between the transitive closure of fuzzy networks and the stationary distribution of stochastic Markov chains to show that the two are equivalent, and we highlight the benefit of using the algebraic representation.

In Secs. 5.1 and 5.2, we introduce notation and background knowledge on the Markov chain and algebraic diffusion model. In Sec. 5.3, we derive the algebraic representation of stochastic Markov chains, demonstrate how to calculate a network's equilibrium distribution algebraically, and establish theorems to show the equivalence. Then in Sec. 6.1 and 6.2 we give two examples: predicting goals in soccer games and calculating the Internet PageRank. Further, we discuss the connection of the algebraic form of stochastic diffusion model to fuzzy set theory [2, 52], and demonstrate the benefits of using the algebraic method by connecting the notion of graph semi-metricity and metric backbone in Secs. 6.1.3 and 6.2.3. By applying the algebraic diffusion model, the semi-metric property of a network can be preserved, and thus it is possible to extract the metric backbone. In the soccer pass networks, the percentage of semi-metric edges has a linear relation to the final game score, while in the Internet hyperlink networks, extracting metric backbones from the graphs after semi-metric edges are removed can accelerate the computation time for PageRank and reduce the storage required for Internet companies.

5.1 Discrete-state Markov Chain

Let $\mathcal{S}$ be a finite or countable state space and P be a transition probability matrix where each entry at position (i, j) is p_{ij}, the probability of the network changing from state i to j, $i, j \in \mathcal{S}$, and $\sum_j p_{ij} = 1$. Let $(X_t : t \in \mathbb{T})$ be a Markov chain

defined on a probability space $(\Omega, \mathcal{F}, \mathcal{P})$, taking values in the state space $\mathcal{S}$ with transition probability matrix P. Then we have

$$\mathbb{P}(X_{t+1} = k | X_t = j, X_{t-1} = x_{t-1}, ..., X_0 = x_0) = \mathbb{P}(X_{t+1} = k | X_t = j) = p_{jk}. \tag{5.1}$$

The Markov chain's memoryless property says that the conditional probability only depends on the states j and k, and not on any earlier states $x_{t-1}, ..., x_0$.

The *equilibrium* or *stationary distribution* of a Markov chain is defined to be a collection of probability distribution of each state $\pi = \{\pi(j) = \sum_{k \in \mathcal{S}} \pi(k) p_{kj} : j, k \in \mathcal{S}\}$.

5.2 Algebraic Diffusion Model

Consider a weighted graph $\mathcal{G}_p(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the collection of all vertices and $\mathcal{E}$ is the collection of all edges in the network. The graph is weighted and each edge weight $w_{ij} \in [0, 1]$ denotes the strength of interaction between node v_i and v_j, $v_i, v_j \in \mathcal{V}$. Consider when each node $v \in \mathcal{V}$ represents a state in $\mathcal{S}$ and $|\mathcal{V}| = |\mathcal{S}|$, we say that $\mathcal{G}_p(\mathcal{V}, \mathcal{E})$ is a graphical representation of state space $\mathcal{S}$. We use $\mathcal{M} = \{\mathcal{G}_p : w_{ij} \in [0, 1], \text{for all } i, j \in \mathcal{S}\}$ to denote the set of graphs with edge weight representing the strength of connection of two nodes and bounded between 0 and 1. Similarly, we use $\mathcal{M}_{sc} = \{\mathcal{G}_p : w_{ij} \in [0, 1] \text{ and } \sum_j w_{ij} = 1, \text{for all } i, j \in \mathcal{S}\}$ to denote the set of graphs that represent Markov chains with w_{ij} being the transition probability from state i to j. It is easy to see that $\mathcal{M}_{sc} \subset \mathcal{M}$.

Isomorphic to $\mathcal{G}_p$, $\mathcal{G}_d(\mathcal{V}, \mathcal{E}')$ captures the association of the network while each edge weight d_{ij} represents how distant or poorly related two nodes v_i and v_j, $v_i, v_j \in \mathcal{V}$ are [2]. The most common way to convert $\mathcal{G}_p(\mathcal{V}, \mathcal{E})$ to $\mathcal{G}_d(\mathcal{V}, \mathcal{E}')$ is to use conversion function φ:

$$d_{ij} = \varphi(w_{ij}) = \frac{1}{w_{ij}} - 1, \quad \forall v_i, v_j \in \mathcal{V}. \tag{5.2}$$

The function $\varphi : [0, 1] \to [0, \infty]$ is a distance function because it is nonnegative, symmetric, and anti-reflexive [2].

5.3 Algebraic Representation of Markov Chain

In this section, we introduce some background knowledge on abstract algebra and derive the algebraic representation of stochastic Markov chains to show that the stationary distribution can be calculated algebraically.

In abstract algebra, an algebraic structure $I = (A, f, g, \dots)$ on a set A is a collection of finitary operations $\{f, g, \dots\}$ on the set itself so that the operations satisfy a list of axioms [53]. A fuzzy graph G is a weighted graph whose adjacency matrix can be represented by a binary relation $R(\mathcal{V}, \mathcal{V})$ such that the Cartesian products of the elements in $\mathcal{V}$ are each assigned a value $r \in [0, 1]$ [52]. In fact, the network of interactions among a set of variables is conceptualized as a fuzzy graph, if all the weights are defined to be in the unit interval $[0, 1]$.

Let us now consider two algebraic structures $I = ([0, 1], DT^1_\vee, DT^1_\wedge)$ and $II = ([0, \infty], f, g)$, where

$$DT^1_\vee = \frac{a + b - 2ab}{1 - ab}, \quad DT^1_\wedge = \frac{ab}{a + b - ab}$$

and

$$f(x, y) = \frac{xy}{x + y}, \quad g(x, y) = x + y.$$

It is shown in [2] that the transitive closure of a fuzzy graph $\mathcal{G}_p$ with algebraic structure I is isomorphic to the distance closure of $\mathcal{G}_p$'s isomorphic graph $\mathcal{G}_d$ using algebraic structure II. This is one example for the generalized n-diffusion process. The distance closure of $\mathcal{G}_d$ with algebraic structure $I = ([0, 1], DT^1_\vee, DT^1_\wedge)$ represents a memoryless n-power diffusion process, and the transitive closure of $\mathcal{G}_p$ with algebraic structure $II = ([0, \infty], f, g)$ represents the n-diffusion with memory, depending on all states before step n.

The stationary distribution of this Markov chain, or the k-power of P (P^k), is obtained by the transition matrix powered by a certain number of steps k until the condition $\pi_{k+1} = \pi_k \cdot P = \pi_k$ is satisfied. This is equivalent to the P^k calculated using $\langle +, \cdot \rangle$ if the pair satisfies the triangular conorm/norm (T-Conorm/Norm) properties [54, 55]. We also know from [2, Thm. 4] that for a given fuzzy graph $G_p = (X, P)$, $G_d = (X, D)$, and the isomorphism φ and Φ, there exists an algebraic structure with $II = ([0, +\infty], f, g)$ with a TD-Conorm/TD-Norm pair $\langle f, g \rangle$ to compute the transitive closure isomorphic distance closure of D, where

$D^{TC} = \Phi(P^{TC})$, and f and g can be computed by:

$$f(d_{ik}, d_{kj}) = \varphi(\varphi^{-1}(d_{ik}) \vee \varphi^{-1}(d_{kj})) \tag{5.3}$$

$$g(d_{ik}, d_{kj}) = \varphi(\varphi^{-1}(d_{ik}) \wedge \varphi^{-1}(d_{kj})). \tag{5.4}$$

Note that although [2, Thm. 4] states the graph G_p is a proximity graph, the result still holds for fuzzy graphs in general [2, p. 232]; G_p does not have to be reflexive, symmetric, or transitive.

Theorem 5. *For a discrete-state Markov chain with state space X, one-time transition probability matrix P, fuzzy graph $\mathcal{G}_p = (X, P)$, a distance graph $\mathcal{G}_d = (X, D)$, and the isomorphism φ and Φ defined, for an algebraic structure $I = ([0,1], \vee, \wedge) = ([0,1], +, \cdot)$ to compute the transitive closure of P, the corresponding algebraic structure $II = ([0, +\infty], f, g)$ to compute the distance closure of D is defined such that:*

$$f(x, y) = \frac{xy - 1}{x + y + 2}, \quad g(x, y) = x + y + xy.$$

Proof. Since each weight w in the fuzzy graph $\mathcal{G}_p$ satisfies the condition that $w \in [0, 1]$, it is straightforward to see that operations $\langle +, \cdot \rangle$ satisfy the four axioms of T-Norm and T-Conorm in the unit interval (commutativity, monotonicity, associativity, and identity) [54, 55]. Now we just need to show that both $+$ and $\cdot$ are binary operations on $[0, 1]^2 \to [0, 1]$. Since every entry in the probability matrix P satisfies $w_{ij} \in [0, 1]$ and $\sum_j w_{ij} = 1$, for all $i, j, k \in S, w_{ij} + w_{kj} \leq 1$ and $w_{ij} \cdot w_{kj} \leq 1$. Combining the two operations, we see that

$$\underbrace{w_{ij} \cdot w_{ki} + \cdots + w_{il} \cdot w_{mi}}_{|\mathcal{S}|} \leq 1, \quad \forall i, j, k, l, m \in \mathcal{S}.$$

With the condition that $\langle +, \cdot \rangle$ is a T-Conorm/T-Norm pair proved, we can apply equations (5.3) and (5.4) to calculate the corresponding functions f and g to calculate the distance closure, where the condition also holds for all integers $n \leq k : D^n = \Phi(P^n)$ [2, Thm. 4]. □

CHAPTER 6

APPLICATIONS OF SEMI-METRICITY IN GRAPHS

The previous chapter shows that a stochastic Markov chain's stationary distribution can be computed algebraically as the corresponding fuzzy graph's k-power of P by applying the n-diffusion model. Note that the TD-Conorm/TD-Norm pair $\langle f, g\rangle$ for the general fuzzy relations case with T-Conorm/T-Norm pair $\langle DT^1_\vee, DT^1_\wedge\rangle$ is similar for the discrete Markov chain with T-Conorm/T-Norm pair $\langle +, \cdot\rangle$. For the general case, function f is the harmonic mean averaged by the total number of paths through two nodes. For the Markov chain case, the measure is a modified version of harmonic mean with penalty:

$$f_{general}(x,y) = \frac{xy}{x+y} = \frac{1}{\frac{1}{x}+\frac{1}{y}} = \frac{HM_g}{n}$$

$$\Downarrow$$

$$f_{stochastic}(x,y) = \frac{xy-1}{x+y+2} = \frac{1}{\frac{1}{x+1}+\frac{1}{y+1}} - 1 = \frac{HM_s}{n},$$

where

$$HM_g(x_1,\ldots,x_n) = \frac{n}{\sum_{i=1}^{n}\frac{1}{x_i}}$$

and

$$HM_s(x_1,\ldots,x_n) = \frac{n}{\sum_{i=1}^{n}\frac{1}{x_i+1}} - n$$

and there is added penalty on the distance measure g:

$$g_{general}(x,y) = x+y$$

$$\Downarrow$$

$$g_{stochastic}(x,y) = x+y+xy.$$

When $x, y \gg 1$, $f_{general} \approx f_{stochastic}$ and $g_{general} \ll g_{stochastic}$; when $x, y \ll 1$, $f_{general} \gg f_{stochastic}$ and $g_{general} \approx g_{stochastic}$. In both cases, the operations $\langle f, g\rangle$

for the general fuzzy graph and the stochastic model are similar. This provides an algebraic explanation for why we can use the original data without fitting to a stochastic model to calculate the diffusion closure for a network and still get a good approximation.

However, we observed that it is more beneficial to study networks using a fuzzy graph than using a stochastic model since fitting the data to a stochastic model can be considered as projecting a high-dimensional surface onto a lower-dimensional space, resulting in loss of information [56]. Since a stochastic model is strictly within the metric space, it does not have properties such as semi-metricity. With semi-metricity existing in a graph, it is possible to calculate the metric backbone of the network, and thus achieve compression [57].

Definition 1 (Semi-metricity)**.** *A distance function d is semi-metric in the algebraic structure $L(Y, \oplus)$ if:*

1. *for all $x \in Y : d(x, x) = 0$;*
2. *for all $x, y \in Y : d(x, y) = d(y, x)$;*
3. *there exist $x, y, z \in Y : d(x, z) \oplus d(z, y) < d(x, y)$.*

Definition 2 (Graph Semi-metricity)**.** *Let $G_p \in \mathcal{M}$ be a fuzzy graph, $\mathcal{M}$ be the set of all fuzzy graphs $\mathcal{G}_p(\mathcal{V}, \mathcal{E})$, and G_d be the isomorphic $\mathcal{G}_d(\mathcal{V}, \mathcal{E}')$ distance graphs such that $G_d = \Phi(G_p)$, defined in the algebraic structures I and II as above, respectively. If for some $n \leq k$, there exist $v_i, v_j \in \mathcal{V} : G_d^n(v_i, v_j) < G_d(v_i, v_j)$, then the edge that connects v_i to v_j, and therefore also the graph, is said to be semi-metric.*

Definition 3 (Metric Backbone)**.** *The metric backbone is the minimum subgraph that preserves the shortest paths of the original weighted network [57–60].*

Metric backbone can be used as a good approximation for a large-scale network by removing the first-order semi-metric edges, achieving compression and hence shortening computation time [57].

6.1 Example: Semi-metricity in Soccer Pass Networks

We obtained pass-by-pass data including the player ids, timestamps, start and end coordinates, outcomes, etc., from [61]. The size of the playing field is normalized

to 100 on each dimension and the field is divided into 32 zones. We also added two states for *shots on target* (33) and *shots off target* (34) for each team. To account for both successful passes and interceptions, we use a network of 4 sub-networks N_{AA}, N_{AB}, N_{BA}, and N_{BB} to represent the successful passes among team A, interceptions of passes from team A by team B (team A fails to pass), interceptions of passes from team B by team A, and successful passes among team B, respectively. The network of networks NN is shown as follows:

$$NN = \begin{bmatrix} N_{AA} & N_{AB} \\ N_{BA} & N_{BB} \end{bmatrix}.$$

All the networks are constructed using the data from the first half of the games and then compared with the results at the end of the games to show the ability of the models to demonstrate trends and team play.

6.1.1 Algebraic Diffusion Model

The fuzzy graph G_p of each sub-network is constructed so that for each team, each edge e_{ij} connecting node v_i and v_j represents that there exist either successful passes or interceptions from zone i to j. The edge weight w_{ij} is proportional to the number of successful passes or interceptions between the two zones that it is connecting, and normalized with a unique linear function so that the maximum edge weight between any two zones is bounded by 1. In this case, we chose function

$$w_{ij}^{norm} = \frac{(1-2\varepsilon)\cdot w_{ij}^{orig} + (2\varepsilon - 1)\cdot \min(w_{ij}^{orig})}{\max(w_{ij}^{orig}) - \min(w_{ij}^{orig})} + \varepsilon, \tag{6.1}$$

where ε is usually set to 0.01 to avoid isolated vertices with edge weights close to the boundary of the original weights. To convert the fuzzy graph G_p to distance graph G_d, we take the nonlinear transform function φ in function (5.2) and apply to each edge weight in G_p. Figure 6.1 illustrates the colormap of a network of networks constructed using data of one match in the Spanish Primera División in season 2012-13.

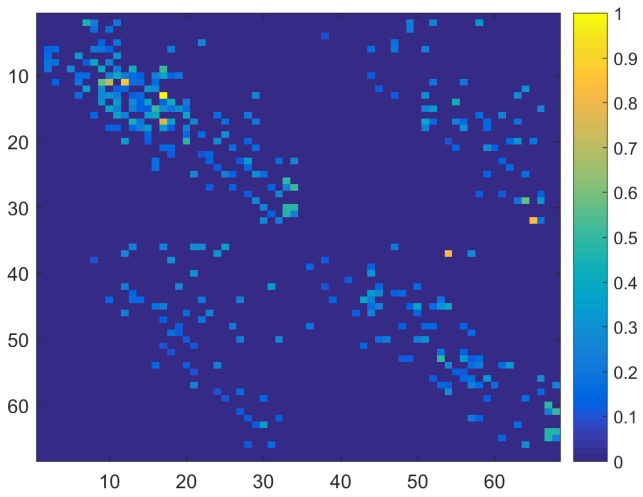


Figure 6.1: Colormap of fuzzy graph of match 1 in the Spanish Primera División in season 2012-13.

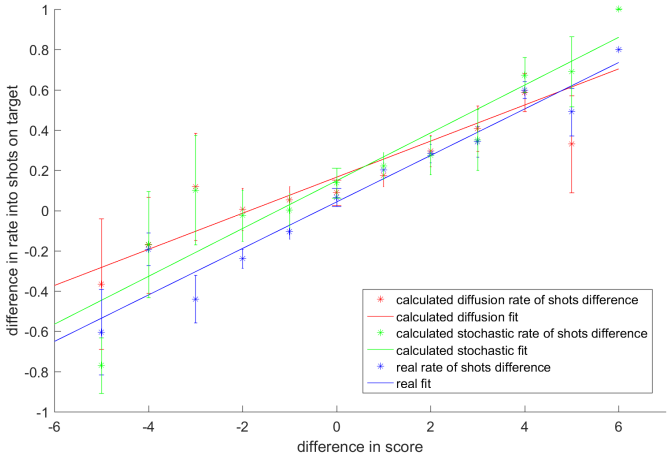
6.1.2 Stochastic Markov Chain

The transition probability matrix P for the stochastic Markov chain is constructed such that each entry p_{ij} is the probability of one team passing or the ball being intercepted by the opposing team from zone i to j, considering the offset for i and j using the structure of network of networks NN. The entries in each row are normalized so that $\sum_j p_{ij} = 1$. The stationary distribution for the game can be found by calculating the nth power of the transition matrix.

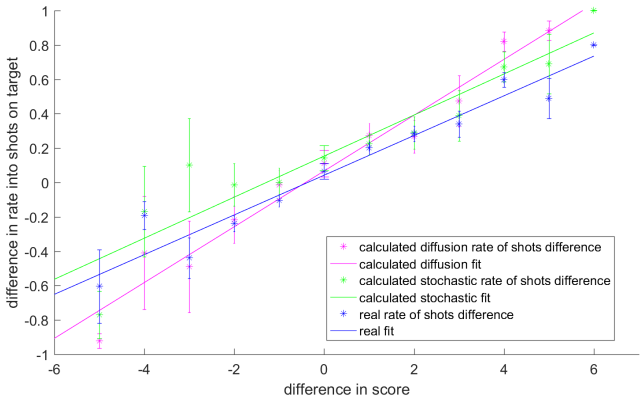
6.1.3 Comparison of Two Models Using Data

Remark. *The n-diffusion G_p^n memoryless and the transitive closure G_p^{TC} of a network stays around the same as a stochastic Markov chain's stationary distribution.*

Figure 6.2 shows that the difference in two teams' *shots on target* stationary distribution calculated with stochastic Markov chain is similar to the difference in two teams' *shots on target* n-diffusion G_p^n calculated with fuzzy and distance network. They are both close to the real measure. Note that Fig. 6.2a shows the diffusion closure result when G_d^k, the powered matrix at step k, is used, while Fig. 6.2b shows the diffusion closure result when the sum ($\oplus \equiv f = min(\cdot)$) of the powered matrix at each step k is used.



(a) n-diffusion = G_d^k

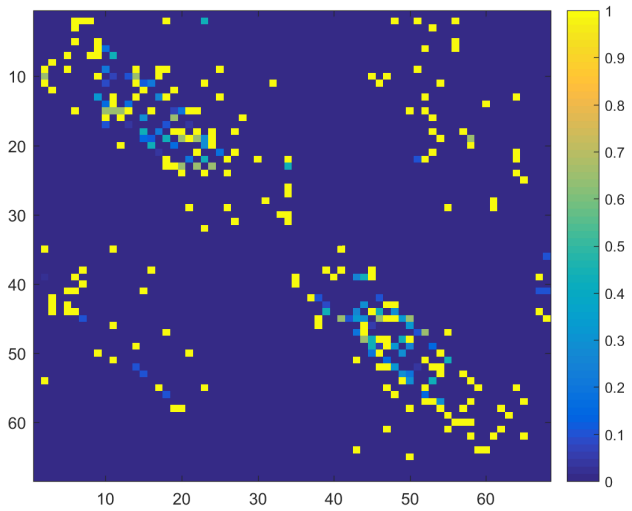


(b) Diffusion closure = $G_d \oplus G_d^2 \oplus \cdots \oplus G_d^k$

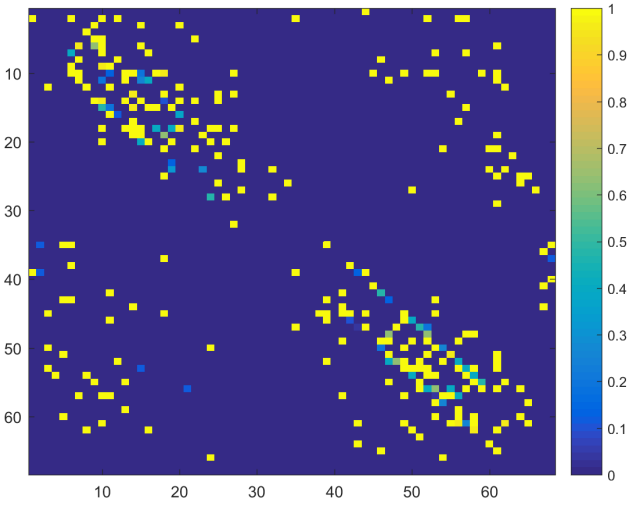
Figure 6.2: Difference in two teams' rates of shots on target with diffusion model's transitive closure and stochastic stationary distribution vs. the score difference of 380 games of the Spanish Primera División in season 2012-13.

6.1.4 Semi-metricity for Trend Prediction

Semi-metricity in the fuzzy graph of the network reveals more critical information of team play, which cannot be captured by the metric graph. Figure 6.3a shows the level of semi-metricity in the pass/interception network using the data of one match, where the upper-left area stands for the home team and the bottom-right area stands for the away team. It is clear that the semi-metricity of the successful passes of the home team is larger than that of the away team. In fact, the home team is also the winner of this specific game. In comparison, Fig. 6.3b shows the same statistics of one game in the German Bundesliga, where the away team is the winner.



(a) Match 1 in the Spanish Primera División in season 2012-13.



(b) Match 9 in the German Bundesliga in season 2012-13.

Figure 6.3: Colormap of semi-metricity of the pass/interception network.

Since the percentage of semi-metric edges is a measure of the degree to which indirect paths have shorter distance than a direct path between two nodes in a network, when the network is spatial soccer passes, we can interpret it as a measure of how flexible one team is at getting the ball from one zone to another. Consider the situation where player x_A on team A tries to pass the ball from zone i to j; however, there are opposing team B players on the direct path $i \to j$. In order to overcome the blockage, the player chooses to first pass the ball to zone k, where there is at least one player y_A on team A. Player y_A, after successfully receiving the ball in zone k, then passes the ball to player z_A in zone j to complete the indirect pass $i \to k \to j$.

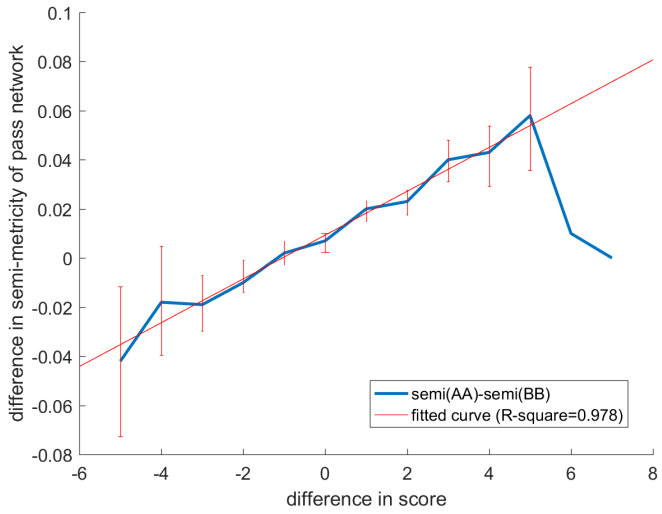


Figure 6.4: Averaged difference in semi-metricity of home and away team vs. their score difference in 1776 matches across 6 soccer leagues.

Hypothesis 1. *When indirect passes are utilized successfully more often than direct ones, the team has higher goal differential.*

In order to verify Hypo. 1, we aggregated the data for 1776 matches across 6 different soccer leagues (Spanish Primera División, German Bundesliga, English Premier League, French Ligue 1, Italian Serie A, and UEFA Europa League). Figure 6.4 illustrates that there exists a linear relationship between the difference in semi-metricity of successful passes and the score difference at the end of the match within some score difference range.

Note that in the high score difference regime ($|\text{score}_A - \text{score}_B| > 5$), the average difference in semi-metricity decreases with the increase of the score difference. This phenomenon might be caused by the fact that when the strengths of two teams differ too much, it is more efficient for the stronger team to attack directly than to go through indirect passes.

Since with stochastic Markov chain model, under the restriction that $\sum_j p_{ij} = 1$, for all $j \in \mathcal{S}$ for each row in the corresponding probability transition matrix P, the weights in the distance network strictly follow the triangle inequality, no semi-metricity would be found when calculating the all-pairs shortest path. Normalizing the probability entry in P can be considered projecting a high-dimensional surface to a lower dimension, which is a lossy process [56]. Thus, it is useful in complex network analysis to utilize more general fuzzy spaces without forcing such constraints, in order to preserve more information from the data.

6.2 Example: Metric Backbone in Internet PageRank Network

PageRank, the probability of arriving at a web page after a large number of clicks, is a measure of the relative importance of web pages according to the directed graph of the web [62]. Similar to a citation network, PageRank is also a diffusion process with damping factor [57]. Consider a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of vertices of n web pages and $\mathcal{E}$ is the set of directed edges e_{ij} which exist if page i has a hyperlink to page j. We then define d_i as the out-degree of node i, representing the number of hyperlinks on page i.

6.2.1 Existing Methods

PageRank x_i assigned to page i is defined as a function of the rank of pages that point to it, divided by the number of links those pages have in total [62] — obtaining in this way a stochastic matrix in $A \in \mathcal{M}_{sc}$:

$$x_i = \sum_{e_{ij} \in \mathcal{E}} d_j^{-1} x_j.$$

In order to guarantee convergence and work around web pages that have no links to other pages, a damping factor is added to the summation term to form the actual model:

$$x_i = (1 - \alpha) + \alpha \sum_{e_{ij} \in \mathcal{E}} d_j^{-1} x_j.$$

The problem can then be formulated as computing the principal eigenvector of a matrix:

$$x = (1 - \alpha) I + \alpha A x,$$

where I is the all-ones vector and $a_{ij} = d_{ij}^{-1}$ if $e_{ij} \in \mathcal{E}$ and zero otherwise.

Intuitively, the algorithm simulates a bored surfer randomly clicking on links to browse one page and another, with probability α to continue at each click. PageRank can be solved both iteratively with power method and algebraically by solving for the steady state distribution of the random walk. The iterative power method is the most straightforward technique. After k iterations, the PageRank of

each web page is:

$$x^{(k+1)} = x + \alpha Ax + \alpha^2 A^2 x^2 + \cdots + \alpha^k A^k x^k.$$

PageRank is not restricted to binary directed graphs. In this work we consider the case of weighted graphs and its PageRank calculation [63, 64].

This problem may be posed as the solution to the a more general linear equation:

$$X = G \cdot X \oplus B,$$

where $G \cdot X = \sum_i^{\oplus} G_{i,k} \otimes X_k$, B specifies the initial conditions, and G a weighted graph or, with proper normalization, a fuzzy graph, $G \in \mathcal{M}$. The PageRank power method of G becomes an approximation of the more general linear equation, because $A \in \mathcal{M}_{sc}$. The solution to the generalized linear equation is:

$$X = (G \oplus G^2 \oplus \cdots \oplus G^k) \otimes B \equiv G^{TC} \otimes B.$$

If we make

$$N \equiv \alpha G$$

the solution becomes:

$$X = (\alpha G \oplus \alpha^2 G^2 \oplus \cdots \oplus \alpha^k G^k) \otimes B \equiv N^{TC} \otimes B,$$

where N^{TC} is the transitive closure. Therefore, the solution to the PageRank problem can be converted to the calculation of the transitive closure and its convergence, if a solution to the general linear equation exists. Here, the parameter α plays an important role, since it guarantees transitive closure convergence to the scaled network $N = \alpha G$, working as a damping factor in G. That is, it is guaranteed that there exists a fixed point to the generalized linear equation.

6.2.2 Metric Backbone Simulation Results

Since PageRank is a diffusion process and the metric backbone contains only the edges that participate in the shortest paths, [57] shows that metric backbone can be used to approximate PageRank accurately and efficiently. The metric backbone can be approximated by removing the first-order semi-metric edges from the orig-

inal graph. They also measured the size reduction of the Neo4j database when the approximate backbone for query evaluation is used to calculate PageRank. The reduction of storage ranges from 16.67% for to 59.14% for highly semi-metric graphs. The authors also conducted experiments on real-life dataset (Facebook, Tuenti, LiveJournal, Twitter, Notredam, and DBLP) on an Amazon EC2 cluster consisting of 16 r3.2xlarge instances, and the highest speedup for PageRank runs close to six times faster for the Tuenti graph. The final result of computing PageRank using the original weighted graph and using the metric backbone results in high Spearman coefficient ranging from 0.76 to 0.98. Using the metric backbones can largely speed up the computation while giving accurate results.

6.2.3 Metric Backbone Algebraic Interpretation

In the work of [57] only simulations were done showing that using the metric backbone in weighted graphs is a useful approximation to PageRank and other algorithms that do not depend directly on the shortest paths (see Sec. 6.2.2). In this section, we want to show algebraically why metric backbone are so good at approximating diffusion processes like PageRank.

In this section, we want to show algebraically why metric backbone results in such good approximation for diffusion processes like PageRank. Recall that the metric backbone is the minimum subgraph that preserves the shortest paths of the original weighted network and is strictly metric with the following properties:

Let G be a fuzzy graph, $G_d = \Phi(G)$ the isomorphic distance graph, and G_{MB} the metric backbone of G. Consider the algebras: $L_1 = ([0,1], \oplus, \otimes)$ with $a \oplus b = a+b$ and $a \otimes b = a \cdot b$, $L_2 = ([0,1], \wedge, \vee)$ with $a \wedge b = \frac{ab}{a+b-ab}$ and $a \vee b = max(a,b)$ and their isomorphic algebras, $L_{1d} = ([0,\infty], f, g)$ with $f(x,y) = \frac{xy-1}{x+y+2}$ and $g(x,y) = x + y + xy$ and $L_{2d} = ([0,+\infty], min, +)$, respectively. Therefore the following is true:

1. $G^{TC_1} \geq G^{TC_2}$ or isomorphically $G_d^{DC_1} \leq G_d^{DC_2}$.
2. $G_{MB}^{TC_2} = G^{TC_2}$ or isomorphically $G_{MB_d}^{DC_2} = G_d^{DC_2}$.
3. $G^{TC_1} \geq G_{MB}^{TC_1} \geq G_{MB}^{TC_2}$ or isomorphically $G_d^{DC_1} \leq G_{MB_d}^{DC_1} \leq G_{MB_d}^{DC_2}$.

In fact it is shown in [2] that:

$$G^{TC_1} \approx G_{MB}^{TC_2}$$

or isomorphically

$$G_d^{DC_1} \approx G_{MB_d}^{DC_2}.$$

Therefore,

$$G^{TC_1} \approx G_{MB}^{TC_1} \geq G_{MB}^{TC_2},$$

and the solution of the general linear equation with algebra L_1 can be approximated by determining the metric backbone in algebra L_2 and applying algebra L_1 to the compressed graph representation, i.e. the metric backbone to calculate the PageRank. This justifies algebraically the results obtained in [57] as described above.

This algebraic result as well as the simulations done in [57] were intuitively expected: since PageRank is seen as a diffusion process in a network, the flow should dynamically travel mainly through metric edges. That is, the metric backbone only contains metric edges leaving out semi-metric edges from the network, by definition. Moreover, from [58] metric edges have edge betweenness greater than zero, i.e. $(b_{i,j} \geq 1)$, and semi-metric edges have betweenness zero, i.e. $(b_{i,j} = 0)$, which explains the diffusion flow going mainly through metric edges, with some diffusion drift through semi-metric edges.

The algebraic interpretation of a diffusion process in this work allows us to better justify the results found in [57].

CHAPTER 7

CONCLUSION

This thesis investigates the effects of changing edges in graphical models through two cases, one with missing edges and one with redundant edges. In this final chapter, we briefly summarize the main results and contributions of the thesis and mention some possible directions for future work.

The first half of the thesis characterizes the performance of message-passing decoders with transiently and permanently missing connections that might be caused by process variation in manufacturing or timing errors in intra-chip communications (or both). We derived density evolution equations to characterize the error probability in the peeling decoder over the BEC and modifications of the Gallager A and Gallager B decoders over the BSC, using erasure symbols to represent missing connections. Although the error probability cannot be driven to 0 in the presence of missing connections, it can be suppressed to a small value η when the channel noise is under a certain decoding threshold ε^*. That is, η-reliable communication is possible with faulty decoders with missing connections. In a sense, even when there exist missing connections in the decoder, equivalent to when the encoder and decoder are effectively using different codebooks, the result is not catastrophic. A novel structural stochastic facilitation is also observed in Gallager B decoders with missing connections.

Future work involves considering not just decoders with missing connections, but also miswired and noisy decoders. It is also interesting to study and model missing connections for min-sum and other belief propagation-like decoders. A natural way to model missing connections when messages are not just a single bit, but are either multi-bits or real-valued, is that the messages are erased due to missing connections. Note that as long as messages are in so-called belief format, all of the symmetry, concentration, and convergence results established in this thesis will hold directly. The performance of decoders with missing connections for finite-length codes is also interesting, especially on the difference between transiently and permanently missing connections, and how to set the parameters of

the codes, change stopping conditions, and modifications to decoding algorithms to take advantage of the stochastic facilitation effect. The error floor region is also of crucial importance for understanding finite-length codes, so it is worthwhile to do more simulations and detailed combinatorial analysis. One may also design new decoder architectures to ensure reliable communication even with miswiring; for example, horizontal connections, a crucial structure in the cortex contributing to the filling in of missing parts in visual images [65, Ch. 8.33], can be added to decoder designs.

Similar scenarios of missing connections in decoders can be also found in data-driven inference tasks, where some constraints or relations among elements are missing. The underlying graphical models can be treated as generalized models for bipartite graphs with missing connections.

The second half of the thesis establishes the equivalence between an algebraic transitive closure of a belief graph and a stochastic Markov chain. The examples of the semi-metric or redundant edges in soccer pass networks and the metric backbone in calculating the internet PageRank demonstrate the benefits of preserving semi-metricity in graph structure to study a network diffusion process. Instead of imposing additional constraints on the original network structure, thus sacrificing topological properties, we may work with the original network structure with different algebras and apply more extended topological methods, such as semi-metricity, to better understand the network topology.

Moreover, the general Markov process (memoryless or with memory) can be seen in light of transitive closure or isomorphically by the diffusion closure, enabling a graphical and geometrical interpretation of a stochastic process. More specifically, a Markov process in the distance space may be described by the equations in Sec. 6.1.3; that is, the Markov chain from one state to another may be interpreted as the harmonic mean of all paths that connect two states (nodes) in a graph with some penalizations. At the same time, using the algebraic representation of the stochastic Markov chain allows a geometrical interpretation of the process. This graphical and geometrical interpretation may be extended to general stochastic processes.

APPENDIX A

PROBABILITY THEORY DEFINITIONS

Before diving into the proof of Thm. 1 in App. B, some probability theory definitions and the Hoeffding-Azuma inequality are reviewed here. Consider a space $(\Omega, \mathcal{F})$, where Ω is a sample space, and a σ-algebra $\mathcal{F}$ contains subsets of Ω. A random variable Z is an $\mathcal{F}$-measurable function from a probability space into the real number. If there is a collection $(Z_\gamma|\gamma \in C)$ of random variables $Z_\gamma : \Omega \to \mathbb{R}$, then

$$\mathcal{Z} = \sigma(Z_\gamma|\gamma \in C)$$

is defined to be the smallest σ-algebra $\mathcal{Z}$ on Ω such that each map $(Z_\gamma|\gamma \in C)$ is $\mathcal{Z}$-measurable.

Definition 4 (Filtration)**.** *Let $\{\mathcal{F}_i\}$ be a sequence of σ-algebras with respect to the same sample space Ω. These $\mathcal{F}_i$ are said to form a* filtration *if $\mathcal{F}_0 \subseteq \mathcal{F}_1 \subseteq \cdots$ are ordered by refinement in the sense that each subset of Ω in $\mathcal{F}_i$ is also in $\mathcal{F}_j$ for $i \leq j$. Also $\mathcal{F}_0 = \{\emptyset, \Omega\}$.*

The conditional expectation of a random variable Z given a σ-algebra $\mathcal{F}$ is a random variable denoted by $E[Z|\mathcal{F}]$.

Definition 5 (Martingale)**.** *Let $\mathcal{F}_0 \subseteq \mathcal{F}_1 \subseteq \cdots$ be a filtration on Ω and let $Z_0, Z_1, \ldots$ be a sequence of random variables on Ω such that Z_i is $\mathcal{F}_i$-measurable. Then $Z_0, Z_1, \ldots$ is a* Martingale *with respect to the filtration $\mathcal{F}_0 \subseteq \mathcal{F}_1 \subseteq \cdots$ if $E[Z_i|\mathcal{F}_{i-1}] = Z_{i-1}$.*

Definition 6 (Doob's Martingale)**.** *Let $\mathcal{F}_0 \subseteq \mathcal{F}_1 \subseteq \cdots$ be a filtration on Ω and let Z be a random variable on Ω. Then the sequence of random variables $Z_0, Z_1, \ldots$ such that $Z_i = E[Z|\mathcal{F}_i]$ is a Doob's Martingale.*

Lemma 3 (Hoeffding-Azuma Inequality [31, 66, 67])**.** *Let $Z_0, Z_1, \ldots$ be a Martingale with respect to the filtration $\mathcal{F}_0 \subseteq \mathcal{F}_1 \subseteq \cdots$ such that for each $i > 0$, the*

following bounded difference condition is satisfied

$$|Z_i - Z_{i-1}| \leq \alpha_i,\ \alpha_i \in [0, \infty).$$

Then for all $n > 0$ *and any* $\xi > 0$*,*

$$\Pr\left[|Z_n - Z_0| \geq \xi\right] \leq 2\exp\left(-\frac{\xi^2}{2\sum_{k=1}^{n}\alpha_k^2}\right).$$

APPENDIX B

CONCENTRATION: PERMANENTLY MISSING CONNECTIONS

The proof of Thm. 1 is an extension of and largely identical to [1, Thm. 2], [31, Thm. 2], or [32, Thm. 4.94]. We want to construct a Doob's Martingale with respect to the fraction of error held on each edge during the random revealing process and to show that the change in the object of interest from one iteration to the next is bounded by a number unrelated to the number of iterations.

Recall that Z denotes the number of incorrect values held at the end of the ℓth iteration for a specific $(g, y, w) \in \Omega$, where g is a specific bipartite Tanner graph to represent the choice of LDPC code with variable node degree $d_{\mathtt{v}}$ and check node degree $d_{\mathtt{c}}$, y is a specific input to the decoder, w is a particular realization of the decoder with missing wires, and Ω is the sample space. Let $\equiv_i$, $0 \leq i \leq m$ be a sequence of equivalence relations on Ω ordered by refinement, such that $(g', y', w') \equiv_i (g'', y'', w'')$ implies $(g', y', w') \equiv_{i-1} (g'', y'', w'')$. The equivalence relations define equivalence classes by partial equalities such that $(g', y', w', u') \equiv_i (g'', y'', w'', u'')$ *if and only if* the realizations of random quantities revealed in the first i steps for both pairs are the same.

Next we use the technique of exposing the edges in the decoding graph in sequence. The first case is when wires are permanently missing. Note that even with positive probability of missing connections α, for a specific code realization, the number of potentially connected edges can be at most $nd_{\mathtt{v}}$. Hence, we expose at most $nd_{\mathtt{v}}$ edges one at a time. At step $i \leq nd_{\mathtt{v}}$, we expose the particular check node socket that is connected to the ith variable node socket. Next, in the following n steps, we expose the received values y_i from the channel one at a time. At the end of the $n(d_{\mathtt{v}} + 1)$ steps, the decoder missing wire probability is also realized, since the defect is permanent. Then we have $(g', y', w') \equiv_i (g'', y'', w'')$ if and only if the information revealed in the first i steps for both pairs is the same.

Now, define $Z_0, Z_1, ..., Z_m$ by

$$Z_i(g, y, w) = E[Z(g', y', w')|(g', y', w') \equiv_i (g, y, w)],$$

where $Z_0 = E[Z]$ and $Z_m = Z$. By construction, $Z_0, Z_1, ..., Z_m$ is a Doob's Martingale. We then use Lem. 3 to give bounds on

$$\Pr[|Z - E[Z]| > nd_{\mathtt{v}}\varepsilon/2] = \Pr[|Z_m - Z_0| > nd_{\mathtt{v}}\varepsilon/2].$$

To use Azuma's inequality, we first need to prove that for each consecutive member in the sequence $Z_0, Z_1, ..., Z_m$, the difference is bounded:

$$|Z_{i+1}(g, y, w) - Z_i(g, y, w)| \leq \delta_i, i = 0, 1, ..., m - 1,$$

where δ_i depends on $d_{\mathtt{v}}, d_{\mathtt{c}}$, and ℓ.

It was shown by Richardson and Urbanke [31] that for the fault-free decoder without any missing wire, when edges are exposed,

$$|Z_{i+1}(g, y, w) - Z_i(g, y, w)| \leq 8(d_{\mathtt{v}}d_{\mathtt{c}})^{\ell}, 0 \leq i \leq nd_{\mathtt{v}}.$$

In our case when there exist permanently missing connections, the difference when exposing edges is that the number of edges existing is smaller, and bounded by $nd_{\mathtt{v}}$. The expected number of edges left is $nd_{\mathtt{v}}(1 - \alpha)$. The bound established above still holds with a change in the step number:

$$|Z_{i+1}(g, y, w) - Z_i(g, y, w)| \leq 8(d_{\mathtt{v}}d_{\mathtt{c}})^{\ell}, 0 \leq i \leq nd_{\mathtt{v}}.$$

It was also shown that when channel outputs are revealed, the difference between consecutive elements in the sequence is bounded by

$$|Z_{i+1}(g, y, w) - Z_i(g, y, w)| \leq 2(d_{\mathtt{v}}d_{\mathtt{c}})^{\ell},$$

where $nd_{\mathtt{v}} \leq i \leq n(d_{\mathtt{v}}+1)$ in the case where some wires are permanently missing. Then the theorem follows from applying Azuma's inequality to the Martingale constructed.

APPENDIX C

CONCENTRATION: TRANSIENTLY MISSING CONNECTIONS

The second case is when wires are transiently missing at each decoding iteration. The Martingale is constructed differently. Instead of exposing edges, at ℓ iterations, we sequentially expose the realization of edges at different iterations. Since each edge can be missing independently from others with probability α, only sockets whose nodes are connected through these edges are affected. In each iteration, there are 2 realizations for each edge (present or missing); then for all previous ℓ iterations, the total number of affected edges is bounded by $2(2d_{\mathsf{v}}d_{\mathsf{c}})^{\ell}$. With symmetry of switching node sockets, the difference between consecutive elements in the sequence is bounded by:

$$|Z_{i+1}(g, y, w) - Z_i(g, y, w)| \leq 8(2d_{\mathsf{v}}d_{\mathsf{c}})^{\ell},$$

where $n(d_{\mathsf{v}} + 1) \leq i \leq m$.

Hence, in the transiently missing wire case, the bounded difference $\delta_i = 8(2d_{\mathsf{v}}d_{\mathsf{c}})^{\ell}$. The theorem follows from applying Azuma's inequality to the Martingale constructed.

REFERENCES

[1] L. R. Varshney, "Performance of LDPC codes under faulty iterative decoding," *IEEE Trans. Inf. Theory*, vol. 57, no. 7, pp. 4427–4444, Jul. 2011.

[2] T. Simas and L. M. Rocha, "Distance closures on complex networks," *Network Sci.*, vol. 3, no. 2, pp. 227–268, Jun. 2015.

[3] R. G. Gallager, *Low-Density Parity-Check Codes*. Cambridge, MA: MIT Press, 1963.

[4] A. A. Al-Yamani, S. Ramsundar, and D. K. Pradhan, "A defect tolerance scheme for nanotechnology circuits," *IEEE Trans. Circuits Syst. I*, vol. 54, no. 11, pp. 2402–2409, Nov. 2007.

[5] S.-L. Jeng, J.-C. Lu, and K. Wang, "A review of reliability research on nanotechnology," *IEEE Trans. Rel.*, vol. 56, no. 3, pp. 401–410, Sep. 2007.

[6] Y. M. Chee and A. C. H. Ling, "Limit on the addressability of fault-tolerant nanowire decoders," *IEEE Trans. Comput.*, vol. 58, no. 1, pp. 60–68, Jan. 2009.

[7] J. R. Heath, P. J. Kuekes, G. S. Snider, and R. S. Williams, "A defect-tolerant computer architecture: Opportunities for nanotechnology," *Science*, vol. 280, no. 5370, pp. 1716–1721, Jun. 1998.

[8] M. Haselman and S. Hauck, "The future of integrated circuits: A survey of nanoelectronics," *Proc. IEEE*, vol. 98, no. 1, pp. 11–38, Jan. 2010.

[9] M. A. Breuer, S. K. Gupta, and T. M. Mak, "Defect and error tolerance in the presence of massive numbers of defects," *IEEE Des. Test. Comput.*, vol. 21, no. 3, pp. 216–227, May-June 2004.

[10] S. Ghosh and K. Roy, "Parameter variation tolerance and error resiliency: New design paradigm for the nanoscale era," *Proc. IEEE*, vol. 98, no. 10, pp. 1718–1751, Oct. 2010.

[11] M. M. Mansour and N. R. Shanbhag, "A 640-Mb/s 2048-bit programmable LDPC decoder chip," *IEEE J. Solid-State Circuits*, vol. 41, no. 3, pp. 684–698, Mar. 2006.

[12] P. Elias, "Computation in the presence of noise," *IBM J. Res. Develop.*, vol. 2, no. 4, pp. 346–353, Oct. 1958.

[13] S. Winograd and J. D. Cowan, *Reliable Computation in the Presence of Noise*. Cambridge, MA: MIT Press, 1963.

[14] S. M. S. Tabatabaei Yazdi, H. Cho, and L. Dolecek, "Gallager B decoder on noisy hardware," *IEEE Trans. Commun.*, vol. 61, no. 5, pp. 1660–1673, May 2013.

[15] S. M. S. Tabatabaei Yazdi, C.-H. Huang, and L. Dolecek, "Optimal design of a Gallager B noisy decoder for irregular LDPC codes," *IEEE Commun. Lett.*, vol. 16, no. 12, pp. 2052–2055, Dec. 2012.

[16] F. Leduc-Primeau and W. J. Gross, "Faulty Gallager-B decoding with optimal message repetition," in *Proc. 50th Annu. Allerton Conf. Commun. Control Comput.*, Oct. 2012, pp. 549–556.

[17] F. Leduc-Primeau, F. R. Kschischang, and W. J. Gross, "Energy optimization of LDPC decoder circuits with timing violations," in *Proc. IEEE Int. Conf. Commun. (ICC 2015)*, Jun. 2015, pp. 412–417.

[18] C.-H. Huang, Y. Li, and L. Dolecek, "Gallager B LDPC decoder with transient and permanent errors," *IEEE Trans. Commun.*, vol. 62, no. 1, pp. 15–28, Jan. 2014.

[19] C. Kameni Ngassa, V. Savin, E. Dupraz, and D. Declercq, "Density evolution and functional threshold for the noisy min-sum decoder," *IEEE Trans. Commun.*, vol. 63, no. 5, pp. 1497–1509, May 2015.

[20] B. Vasić and S. K. Chilappagari, "An information theoretical framework for analysis and design of nanoscale fault-tolerant memories based on low-density parity-check codes," *IEEE Trans. Circuits Syst. I*, vol. 54, no. 11, pp. 2438–2446, Nov. 2007.

[21] E. Dupraz, D. Declercq, B. Vasić, and V. Savin, "Analysis and design of finite alphabet iterative decoders robust to faulty hardware," *IEEE Trans. Commun.*, vol. 63, no. 8, pp. 2797–2809, Aug. 2015.

[22] O. A. Rasheed, P. Ivaniš, and B. Vasić, "Fault-tolerant probabilistic gradient-descent bit flipping decoder," *IEEE Commun. Lett.*, vol. 18, no. 9, pp. 1487–1490, Sep. 2014.

[23] B. Vasić, P. Ivaniš, S. Brkic, and V. Ravanmehr, "Fault-resilient decoders and memories made of unreliable components," in *Proc. 2015 Inf. Theory Appl. Workshop*, Feb. 2015.

[24] L. R. Varshney, "Toward limits of constructing reliable memories from unreliable components," in *Proc. IEEE Inf. Theory Workshop (ITW'15)*, Oct. 2015, pp. 114–118.

[25] S. Brkic, P. Ivaniš, and B. Vasić, "Guaranteed error correction of faulty bit-flipping decoders under data-dependent gate failures," in *Proc. 2016 IEEE Int. Symp. Inf. Theory*, Jul. 2016, pp. 1561–1565.

[26] C.-H. Huang, Y. Li, and L. Dolecek, "Belief propagation algorithms on noisy hardware," *IEEE Trans. Commun.*, vol. 63, no. 1, pp. 11–24, Jan. 2015.

[27] A. Karbasi, A. H. Salavati, A. Shokrollahi, and L. R. Varshney, "Noise facilitation in associative memories of exponential capacity," *Neural Comput.*, vol. 26, no. 11, pp. 2493–2526, Nov. 2014.

[28] S. Brkic, O. A. Rasheed, P. Ivaniš, and B. Vasić, "On fault-tolerance of the Gallager B decoder under data-dependent gate failures," *IEEE Commun. Lett.*, vol. 19, no. 8, pp. 1299–1302, Aug. 2015.

[29] H. Chen, L. R. Varshney, and P. K. Varshney, "Noise-enhanced information systems," *Proc. IEEE*, vol. 102, no. 10, pp. 1607–1621, Oct. 2014.

[30] P. Ivaniš, B. Vasić, and D. Declercq, "Performance evaluation of faulty iterative decoders using absorbing Markov chains," in *Proc. 2016 IEEE Int. Symp. Inf. Theory*, Jul. 2016, pp. 1566–1570.

[31] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 599–618, Feb. 2001.

[32] T. Richardson and R. L. Urbanke, *Modern Coding Theory*. Cambridge: Cambridge University Press, 2008.

[33] H. Mori and T. Wadayama, "Performance analysis based on density evolution on fault erasure belief propagation decoder," in *Proc. 2016 IEEE Int. Symp. Inf. Theory*, Jul. 2016, pp. 1571–1575.

[34] L. Bazzi, T. J. Richardson, and R. L. Urbanke, "Exact thresholds and optimal codes for the binary-symmetric channel and Gallager's decoding algorithm A," *IEEE Trans. Inf. Theory*, vol. 50, no. 9, pp. 2010–2021, Sep. 2004.

[35] V. Guruswami, "Iterative decoding of low-density parity check codes (a survey)," arXiv:cs/0610022 [cs.IT]., 2006.

[36] M. J. Wainwright, "Estimating the "wrong" graphical model: Benefits in the computation-limited setting," *J. Mach. Learn. Res.*, vol. 7, pp. 1829–1859, Dec. 2006.

[37] T. J. Hamilton, S. Afshar, A. van Schaik, and J. Tapson, "Stochastic electronics: A neuro-inspired design paradigm for integrated circuits," *Proc. IEEE*, vol. 102, no. 5, pp. 843–859, May 2014.

[38] A. Amraoui, A. Montanari, T. Richardson, and R. L. Urbanke, "Finite-length scaling for iteratively decoded LDPC ensembles," *IEEE Trans. Inf. Theory*, vol. 55, no. 2, pp. 473–498, Feb. 2009.

[39] M. A. Breuer and H. Zhu, "An illustrated methodology for analysis of error tolerance," *IEEE Des. Test. Comput.*, vol. 25, no. 2, pp. 168–177, Mar.-Apr. 2008.

[40] Z. Jiang and S. K. Gupta, "Threshold testing: Improving yield for nanoscale VLSI," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 28, no. 12, pp. 1883–1895, Dec. 2009.

[41] K. Flamm, "The impact of DRAM design innovation on manufacturing profitability," *Future Fab Int.*, no. 35, Nov. 2010.

[42] S. Mittal, "A survey of architectural techniques for managing process variation," *ACM Comput. Surv.*, vol. 48, no. 54, May 2016.

[43] A. Barrat, M. Barthélemy, R. Pastor-Satorras, and A. Vespignani, "The architecture of complex weighted networks," *Proc. Natl. Acad. Sci. U.S.A.*, vol. 101, no. 11, pp. 3747–3752, Jan. 2004.

[44] A. Arenas, A. Daz-Guilera, J. Kurths, Y. Moreno, and C. Zhou, "Synchronization in complex networks," *Phys. Rev.*, vol. 469, no. 3, pp. 93–153, 2008.

[45] V. Latora and M. Marchiori, "Economic small-world behavior in weighted networks," *Eur. Phys. J. B*, vol. 32, no. 2, pp. 249–263, 2003.

[46] M. E. J. Newman, "Scientific collaboration networks. II. shortest paths, weighted networks, and centrality," *Phys. Rev. E*, vol. 64, no. 1, pp. 016 133–1 – 016 133–7, Jun. 2001.

[47] ——, "Analysis of weighted networks," *Phys. Rev. E*, vol. 70, no. 5, pp. 056 131–1 – 056 131–9, Nov. 2004.

[48] D. Huang, "Synchronization in adaptive weighted networks," *Phys. Rev. E*, vol. 74, no. 4, pp. 046 208–1 – 046 208–5, Oct. 2006.

[49] M. Chavez, D.-U. Hwang, A. Amann, H. G. E. Hentschel, and S. Boccaletti, "Synchronization is enhanced in weighted complex networks," *Phys. Rev. Lett.*, vol. 94, no. 21, pp. 218 701–1 –218 701–4, Jun. 2005.

[50] P. Pons and M. Latapy, "Computing communities in large networks using random walks," in *Proc. 20th Int. Symp. Comput. and Inf. Sci. (ISCIS 2005)*, vol. 3733. Springer Berlin Heidelberg, Oct. 2005, pp. 284–293.

[51] P. Blanchard and D. Volchenkov, *Random Walks and Diffusions on Graphs and Databases: An Introduction*, 1st ed. Springer, 2011.

[52] D. P. Filev, "Klir and Yuan's fuzzy sets and fuzzy logic," *J. Intell. Fuzzy Syst.*, vol. 4, no. 2, pp. 175–176, Mar. 1996.

[53] P. M. Cohn, *Universal Algebra*. New York, NY: Harper & Row, 1965.

[54] K. Menger, "Statistical metrics," *Proc. Natl. Acad. Sci. U.S.A.*, vol. 28, no. 12, pp. 535–537, Dec. 1942.

[55] B. Schweizer and A. Sklar, *Probabilistic Metric Spaces*. New York, NY: North-Holland, 1983.

[56] T. Simas, M. Chavez, P. Rodriguez, and A. Diaz-Guilera, "An algebraic topological method for multimodal brain networks comparisons," *Front. Psychol.*, vol. 6, no. 904, 2015.

[57] V. Kalavri, T. Simas, and D. Logothetis, "The shortest path is not always a straight line: Leveraging semi-metricity in graph analysis," *Proc. VLDB Endowment*, vol. 9, no. 9, pp. 672–683, May 2016.

[58] T. M. L. M. De Simas, "Stochastic models and transitivity in complex networks," Ph.D. thesis, Indiana University, Bloomington, IN, May 2012.

[59] T. Simas, S. Chattopadhyay, C. Hagan, P. Kundu, A. Patel, R. Holt, D. Floris, J. Graham, C. Ooi, R. Tait, M. Spencer, S. Baron-Cohen, B. Sahakian, E. Bullmore, I. Goodyer, and J. Suckling, "Semi-metric topology of the human connectome: Sensitivity and specificity to autism and major depressive disorder," *PLoS ONE*, vol. 10, no. 8, pp. 1–20, Aug. 2015.

[60] T. Simas and J. Suckling, "Commentary: Semi-metric topology of the human connectome: Sensitivity and specificity to autism and major depressive disorder," *Front. Neurosci.*, vol. 10, no. 353, Aug. 2016.

[61] L. Gyarmati, H. Kwak, and P. Rodriguez, "Searching for a unique style in soccer," in *Proc. KDD Workshop on Large-Scale Sports Analytics*, Aug. 2014.

[62] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank citation ranking: Bringing order to the web," Stanford InfoLab, Tech. Rep. SIDL-WP-1999-0120, 1999.

[63] S. Fortunato and A. Flammini, "Random walks on directed networks: the case of PageRank," *Int. J. Bifurc. Chaos*, vol. 17, no. 7, pp. 2343–2353, Jul. 2007.

[64] M. R. Meiss, F. Menczer, S. Fortunato, A. Flammini, and A. Vespignani, "Ranking web sites with real user traffic," in *Proc. 2008 Int. Conf. Web Search and Data Mining (WSDM'08)*, Feb. 2008, pp. 65–76.

[65] P. Brodal, *The Central Nervous System*. New York, NY: Oxford University Press, 2016.

[66] K. Azuma, "Weighted sums of certain dependent random variables," *Tohoku Math. J.*, vol. 19, no. 3, pp. 357–367, 1967.

[67] W. Hoeffding, "Probability inequalities for sums of bounded random variables," *J. Am. Stat. Assoc.*, vol. 58, no. 301, pp. 13–30, Mar. 1963.