

UNIVERSITY OF CALIFORNIA
RIVERSIDE

Information Gathering on Bounded Degree Trees and Properties of Random
Matrices

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Mathematics

by

Parker Richard Williams

June 2017

Dissertation Committee:

Professor Kevin Costello, Chairperson
Professor Mei Chu Chang
Professor Amir Moradifam

Copyright by
Parker Richard Williams
2017

The Dissertation of Parker Richard Williams is approved:

Committee Chairperson

University of California, Riverside

Acknowledgements

I am deeply indebted to Professors Arnott, Chrobak and Costello for providing me with such rich problems to spend my graduate career grappling with. I was very fortunate to use my mathematical training in so many fields, and with their guidance I matured greatly as a researcher. I am further indebted to my advisor, Professor Costello for his mentorship, patience, and encouragement. His excellent presentation of the probabilistic method was delightful and inspiring, I have no doubt that Erdős would applaud. I would like to thank the UCR Mathematics Department for providing me with the resources and support to conduct my research. I also owe a great deal of gratitude to Professor Chari for guiding me professionally and leading our outreach programs, encouraging me to take the lead on several projects. I must also deeply thank Professor Moradifam and Professor Chang for their roles in my committee and enduring my drafts and talks. Lastly I owe great thanks to the University of California for supporting my research and studies throughout the last few years.

Dedication

To all my friends and family, I could not have done this without you. To all my good friends in the mathematics department, I thank you for your wonderful camaraderie.

ABSTRACT OF THE DISSERTATION

Information Gathering on Bounded Degree Trees and Properties of Random Matrices

by

Parker Richard Williams

Doctor of Philosophy, Graduate Program in Mathematics
University of California, Riverside, June 2017
Professor Kevin Costello, Chairperson

In this thesis two questions are addressed. First is the question of the time complexity of an algorithm gathering information on a tree with unknown topology. The tree is known to have maximum depth D and maximum degree Δ . This has been an open problem since being asked by Chlamtac and Kutten in 1985. This thesis resolves the question with upper and lower bounds that are essentially tight. The second question this dissertation addresses is what proportion of graphs have a fixed type of spectrum. We show that at most a $2^{-cn^{3/2}}$ proportion of graphs on n vertices have integral spectrum. This improves on previous results of Ahmadi, Alon, Blake, and Shparlinski, who in 2009 showed that the proportion of such graphs was exponentially small.

Contents

1	Introduction	1
1.1	Previous Work	2
2	Graph Theory	4
2.1	Properties of Graphs and Trees	4
3	Random Graphs and Matrices	9
3.1	Properties of Random Adjacency Matrices	9
4	The Ad-Hoc Radio Network Model	13
4.1	Description of the Model	13
4.2	Essential Features of the Model	14
5	Tools from Extremal Set Theory	18
5.1	Strongly Selective Families of Sets	19
6	Gathering on Degree Three Trees	26
6.1	The Degree 3 Tree Algorithm	26
7	Gathering on Bounded Degree Trees	32
7.1	Degree Δ with Rounds	32
7.2	Degree Δ Without Rounds	39
7.3	Lower Bounds for Complete Tree	50
8	On the Number of Integral Graphs	56
8.1	Deterministic properties of adjacency matrices	57
8.2	The Proof of Theorem 8.0.2	60
9	Conclusions	65
9.1	Information Gathering in Ad-Hoc Radio Networks having Maximum Degree Δ	65
9.2	Matrices Having Spectra of Fixed Type	66
	Bibliography	68

Chapter 1

Introduction

Given any task, it is natural to ask “how long might this task take”? Analysis of the time complexity of algorithms is as old as computer science itself. Many unanswered questions occupy this space, and the *gossiping problem* on a graph has been well studied for 32 years. The gossiping problem asks in a graph where each vertex has a rumor, and in each time step a vertex in the graph may share all rumors it knows with its neighbors. The problem then asks how many time steps are needed to have all vertices know all rumors, regardless of the structure of the graph? In this thesis we will focus on the *gathering problem* on trees with unknown topology. We will resolve a question asked by Chrobak, Costello, Gasieniec, Kowalski in [6] which is a modification of the gathering problem. In this modified version the maximum number of children any vertex in a tree \mathcal{T} is Δ , and the maximum depth D are known.

Another question addressed by this thesis is given a graph G , there is an associated matrix of G called the adjacency matrix which we will denote $A(G)$, where $A_{i,j}$ is 1 if

vertex i is connected to vertex j and $A_{i,j}$ is 0 otherwise. The spectra of these matrices are well-studied and encode many facts about the graph that are difficult to compute directly. This thesis answers a question regarding the problem “how many graphs have a spectrum of a fixed type”. The type we consider is when the eigenvalues of $A(G)$ are all integers. We offer an improvement on known bounds but suspect our bound is still far from tight.

1.1 Previous Work

The first question was asked in its current form in a paper by Bar-Yehuda, Goldreich, and Itai in [2], which at the time was the gossiping problem on graphs but with the added essential feature of *collision*. The question had existed in looser forms since the 1960s regarding various toy combinatorial problems about gossiping on telephone networks made popular by Erdős. Since the work of Bar-Yehuda, Goldreich, and Itai added the essential feature of collision, much progress has been made and the two “primitives” have been distilled out of all questions regarding algorithms for the dissemination of information on graphs with unknown structure. It is worth noting that in the literature this problem has also been called ad hoc networks with unknown topology. The two primitives are the following:

- (1) The *broadcasting problem*, one to all dissemination.
- (2) The *gossiping problem*, all to all dissemination.

One way of approaching the latter is in two phases; *gathering* and *broadcasting*. First, gather all rumors to a “town crier” vertex in the graph, and then let the “crier” broadcast to the

rest. De Marco demonstrated in [16] a bound that broadcasting was much faster than any known bound for gathering, so in the current landscape, any improvement in gathering is an improvement in gossiping. This thesis considers then the problem of gathering on a tree with unknown structure, with only the information that the height of the tree is bounded by D and no vertex may have more than Δ children, which was a question raised in [6].

The second question this thesis investigates is another form of “primitive.” It is an unresolved question in random matrix theory and random graph theory. An extremely well-studied question asks given, A_n a random $n \times n$ matrix, what can one say about the limiting behavior of properties of $\lim_{n \rightarrow \infty} A_n$?

Spectral graph theory is an extremely active field of research, the study of the eigenvalues of the various matrix representations of graphs. Integral graphs were studied first by Harary and Schwenk who had described the task of classifying all integral graphs as “intractable.” This thesis investigates the probability that a graph chosen at random is integral. The first non-trivial bound for this question was found in 2009 by Ahmadi, Alon, Blake, and Shparlinski to be $2^{-n/400}$ where n is the number of vertices of the graph. These graphs appear to be building blocks in spectral graph theory, as well as model perfect state transfer in quantum networks as found by Christandl et al. and Bašić et al. ([5, 3]).

Chapter 2

Graph Theory

2.1 Properties of Graphs and Trees

The underlying combinatorial structure we will consider for our algorithms is a directed graph. In this thesis we will wish to consider whether or not two vertices can send or receive messages from each other and this pairwise relationship is easily captured through a graph.

Throughout this thesis we will write $[n]$ to mean $1, \dots, n$. At times floors and ceilings will be omitted for clarity.

Definition 2.1.1. A *graph* G is comprised of two sets, V, E where $E \subset V \times V$.

The model this thesis works with is thinking of graphs as networks with vertices being transmitters, and a line between two vertices indicating the possibility of transmission.

Definition 2.1.2. The *degree* of a vertex v is the number of edges incident to the vertex v .

Definition 2.1.3. In a graph G with $x, y \in V$ a *path* P from x to y is a non-repeating sequence of vertices starting at x and ending at y written $P = \{x, v_1 \dots v_k, y\}$ where every sequential pair of vertices in P is in E .

Definition 2.1.4. A graph G is called *connected* if for any $v_i, v_j \in V$ there exists a path from v_i to v_j .

Definition 2.1.5. In a graph G a *cycle* C is a path where the first and last element of C are the same vertex.

Definition 2.1.6. A *tree* \mathcal{T} is a connected graph G that has no cycles.

This construction is the primary focus of this thesis and throughout we will develop tools to discuss properties of trees.

An equivalent definition is the following:

Definition 2.1.7. A graph G is a tree if and only if there is a unique path between any two vertices.

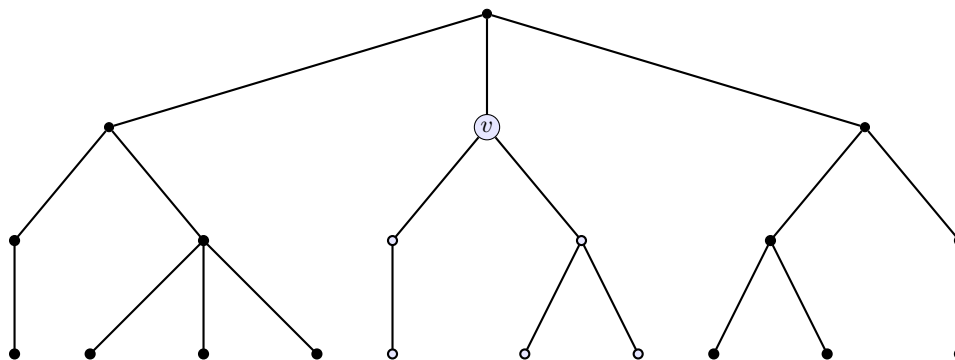
Definition 2.1.8. In a tree \mathcal{T} , a vertex is distinguished as the *root* of the tree and will be labeled r .

Given a vertex $v \in \mathcal{T}$, we will wish to discuss its local structure. With a root $r \in \mathcal{T}$ designated, we have a natural way to do this. For every vertex $v \in \mathcal{T}$, consider the path to the root r . Call it $P_{v,r} = \{v, u_1, \dots, u_m, r\}$ which then gives us another notion to help us discuss the local structure around v and, luckily, as observed by 2.1.7, it is unique.

Definition 2.1.9. Let $P_{w,r}$ be a path as above, where $P = \{w, v, u_1, \dots, u_m, r\}$. We will call v the *parent* of w , and w the *child* of v .

Definition 2.1.10. The *tree rooted at v* is the collection of all vertices w so that for the path $P_{w,r}$ we have $v \in P_{w,r}$ and we will write it as \mathcal{T}_v .

Example 2.1.11. A tree with a subtree rooted at v , the subtree indicated by lighter vertices.



Definition 2.1.12. Let W be the collection of vertices $W \subset V$ where for every $\tilde{w} \in W$ the vertex v is the first element of $P_{\tilde{w},r}$ to follow \tilde{w} . We will call the set of vertices W *siblings*.

On a given tree \mathcal{T} with root r we now have an ability to talk about the distance a vertex $v \in \mathcal{T}$ is from r , which in terms of the model of transmission is the number of “hops” to the root.

Definition 2.1.13. For all vertices $v \in \mathcal{T}$ the depth of a vertex is the length of the unique path P connecting v to r .

Remark 2.1.14. In a graph, the maximum degree of a vertex v is the number of vertices incident to v . However, in the setting of a tree, the term degree and number of children are used equivalently. We will say that a tree with maximum degree Δ means that the maximum number of children of any vertex is Δ .

Definition 2.1.15. Let \mathcal{T} be a tree where every vertex $v \in \mathcal{T}$ where v is not a leaf has exactly k children, and that every leaf has the same distance to the root. We will call \mathcal{T} k -regular, or the complete k -ary tree.

A notion introduced in [6] is a variation on the definition of height, which can help capture how “bushy” a tree is.

Definition 2.1.16. Let v be a vertex and let v_1, \dots, v_l be the children of v . For any integer $\gamma = 1, \dots, n - 1$ we define the γ -height of any vertex $v \in \mathcal{T}$ recursively, and will write it as $h_\gamma(v)$. If v is a leaf, then $h_\gamma(v) = 0$. Let c be the maximum γ -height of any of the children of v . If v has at least γ children with γ -height c , then $h_\gamma(v) = c + 1$. Otherwise $h_\gamma(v) = c$.

We will call the γ depth of a tree $h_\gamma(\mathcal{T})$, the γ depth of the root r so $h_\gamma(\mathcal{T}) = h_\gamma(r)$.

When $\gamma = 1$, we will call $h_1(v)$ the height of the vertex v . We are now ready to present some important lemmas, the first of which appears in [6].

Lemma 2.1.17. Let \mathcal{T} be a tree on n vertices then $\forall \gamma \in \mathbb{N}$

$$h_\gamma(\mathcal{T}) \leq \log_\gamma(n).$$

Proof. Let \mathcal{T}_v be the tree rooted at v . It suffices to show that $|\mathcal{T}_v| \geq \gamma^{h_\gamma(v)}$ for all $v \in \mathcal{T}$.

We induct on the height of v . If v is a leaf, it has height 0, and the inequality trivially holds. If v is not a leaf then v has γ -height z , or $h_\gamma(v) = z$. If v has a child with height w , then by induction $|\mathcal{T}_v| \geq |\mathcal{T}_w| \geq \gamma^z$. If all children of v have gamma height smaller than z then v must have at least γ children all having γ -height $z - 1$. So by induction we get $|\mathcal{T}_v| \geq \gamma \cdot \gamma^{z-1} = \gamma^z$. □

Lemma 2.1.18. If v has children $v_1 \dots v_l$ then for $i \in [l]$, any collection of $I \subset v_1 \dots v_l$ with $|I| = i$ some child $\tilde{v} \in I$ must satisfy $h_i(v) - h_i(\tilde{v}) \geq 1$.

Proof. Fix an integer $i \in [l]$. Let $I \subset \{v_1 \dots v_l\}$ be a collection of vertices with common parent v let $c = \min_{j \in I} h_i(v_j)$. This means v has i children with i -height at least c , thus $h_i(v) \geq c + 1$. □

Lemma 2.1.19. If v is the parent of \tilde{v} then $h_\gamma(v) \geq h_\gamma(\tilde{v})$ for all γ .

Proof. From the recursive definition of 2.1.16 we can see that $h_\gamma(\cdot)$ increases monotonically from child to parent. □

Chapter 3

Random Graphs and Matrices

3.1 Properties of Random Adjacency Matrices

A well-known and frequently used fact about graphs is that for every graph G one may associate an adjacency matrix $A(G)$. Taking the numbers $1, \dots, n$ to enumerate the vertices of G , the $A_{i,j}$ entry of A be 1 if vertex i is connected to vertex j and 0 otherwise.

First we see that A is necessarily a symmetric matrix since if vertex i is connected to vertex j , then by definition j is connected to vertex i hence $A_{i,j} = A_{j,i}$.

The connection of random matrices and random graphs is immediate. If G is a random matrix on n vertices with each edge being included with probability p , this could also be considered a random symmetric matrix A with $A_{i,j}$ independent when $i \geq j$, where $P(A_{i,j} = 1) = p$ and $P(A_{i,j} = 0) = 1 - p$. While this immediately produces a random $\{0, 1\}$ matrix, many theorems this thesis will make use of do not require that the entries be 0 or 1.

A central result in random matrix theory that still is quite important in the restricted $\{0, 1\}$ case is a result due to Wigner [21] that helps us understand the limiting behavior of A as the size of the matrix goes to infinity. This can equivalently be thought of as the limiting behavior of the eigenvalues of the adjacency matrix of a graph G as the number of vertices goes to infinity.

Theorem 3.1.1. (Wigner’s Semicircle Law) Let A be a random symmetric matrix whose (i, j) entry is $A_{i,j}$, a real-valued random variable satisfying the following properties:

- $A_{i,j}$ independent when $i \leq j$
- The distribution of $A_{i,j}$ is symmetric about 0
- The variance of the distribution $A_{i,j}$ is 1
- For each $k \geq 2$, the k^{th} moment of the distribution of $A_{i,j}$ is uniformly bounded by a constant C_k

Let A_n be a random matrix with size $n \times n$ with entries $A_{i,j}$. For real numbers α, β , let $E_n(\alpha, \beta)$ be the number of eigenvalues of A_n lying between $\alpha\sqrt{n}$ and $\beta\sqrt{n}$. Then for any α, β with probability 1 we have

$$\lim_{n \rightarrow \infty} \frac{E_n(\alpha, \beta)}{n} = \int_{\alpha}^{\beta} \sqrt{4 - x^2} dx.$$

Definition 3.1.2. Let the *spectrum* of G be the collection of eigenvalues of $A(G)$.

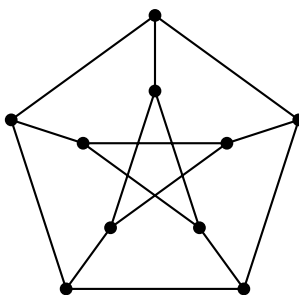
If G has n vertices and since the matrix is symmetric with dimension n , the eigenvalues are real, and we will list them from smallest to largest as $\lambda_1, \dots, \lambda_n$. This thesis will focus on graphs having spectrums of a specific type.

Definition 3.1.3. Call a graph G *integral* if all the eigenvalues of $A(G)$ are integers.

Examples of such graphs are K_n , the complete graph on n vertices, $K_{n,n}$, the complete bipartite graph on $2n$ vertices, and the Payley graph on q vertices where q is a odd square prime power. More recently, integral graphs have been seen as useful in designing quantum spin networks with perfect state transfer.

Example 3.1.4. The Kneser graph $KG_{5,2}$, is integral with spectrum:

$$\{3, 1, 1, 1, 1, 1, -2, -2, -2, -2\}$$



In this thesis we will wish to discuss the properties of the spectrum of a matrix. Let A be an $n \times n$ matrix. We will call A_k the upper left $k \times k$ minor of A .

Definition 3.1.5. Let $\text{Mult}(A, \lambda)$ be the multiplicity of λ as an eigenvalue of A .

As we have seen earlier, we are interested in the limiting behavior of these distributions and one property that will prove very useful to us is the interaction spectrum of A_k and A_{k+1} . The following theorem is a well-known result. This is sometimes called the *interlacing eigenvalues theorem for bordered matrices* also sometimes called the *Cauchy Interlacing Theorem*. It is originally stated for Hermitian matrices and is presented in [13] but will be stated for the purposes of this thesis in terms of adjacency matrices.

Theorem 3.1.6. Let A be a symmetric $n \times n$ $\{0, 1\}$ valued matrix and let x be a vector in $\{0, 1\}^n$. Let A_{n+1} be the matrix obtained by adding x in the following way

$$A_{n+1} = \begin{pmatrix} A_n & x \\ x^T & 0 \end{pmatrix}.$$

Let the eigenvalues of A be written in increasing order as $\lambda_1, \dots, \lambda_n$ and the eigenvalues of A_{n+1} be written in increasing order as $\hat{\lambda}_1, \dots, \hat{\lambda}_{n+1}$. Then

$$\hat{\lambda}_1 \leq \lambda_1 \leq \hat{\lambda}_2 \leq \dots \leq \lambda_{n-1} \leq \hat{\lambda}_n \leq \lambda_n \leq \hat{\lambda}_{n+1}.$$

This gives us a discrete continuity result for how the multiplicity of eigenvalues change as the size of a matrix grows by 1, or equivalently how the spectrum of a random graph changes as a new vertex is added.

Corollary 3.1.7. Let A_n and A_{n+1} be symmetric random $\{0, 1\}$ matrices as above where A_{n+1} is obtained by adding a single vector $x \in \{0, 1\}^n$. Then

$$|\text{Mult}(A_n, \lambda) - \text{Mult}(A_{n+1}, \lambda)| \leq 1.$$

Chapter 4

The Ad-Hoc Radio Network Model

4.1 Description of the Model

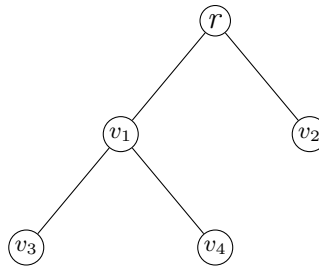
We will consider the question of information traveling through the nodes of a network. We will represent nodes of this network as vertices of a tree. Given a tree \mathcal{T} on n vertices with maximum degree Δ , each vertex has a rumor that it wants to get to the root of the tree r . Vertices are labeled $1, \dots, n$ and each vertex knows its own label. We discretize time, and at each time step t , a protocol must instruct a vertex $v \in \mathcal{T}$ to transmit its rumor or any information to its parent or remain silent. This decision can be based on the time t , the label of the vertex $i \in [n]$, or any rumors or other information transmitted to v before t . This set of decisions however may only rely upon the topology of \mathcal{T} that the protocol has learned up until this time. When a vertex $v \in \mathcal{T}$ transmits, there are two scenarios that can occur:

- (1) At time t , v transmits and no sibling of v transmits. The parent of v receives all information sent by v .
- (2) At time t , if v and w are siblings and both transmit, the common parent of v, w receives no rumor.

We will call the first outcome a *success* and we will call the second outcome a *collision*.

Importantly, a vertex is unaware of the outcome of a transmission.

Example 4.1.1. Let us now see how a simple protocol functions on a small tree



Time	Transmitting vertices	Result
1	v_3	v_1 receives rumor from v_3
2	v_3, v_4	v_3, v_4 collide, v_1 receives no rumors
3	v_2, v_4	v_1, r receive rumors from v_4, v_2 respectively
4	v_1, v_2	v_1, v_2 collide, r receives no rumors
5	v_1	r receives rumor for v_1 which includes rumors from v_3, v_4
6	v_2	r receives rumor for v_2 and the protocol finishes

4.2 Essential Features of the Model

We now need to more carefully define what we needed for that example. Let \mathcal{T} be a tree and fix $v \in V(\mathcal{T})$. Let v_1, \dots, v_l be children of v .

Definition 4.2.1. A *rumor* is information held by a vertex. In a tree \mathcal{T} , each vertex $v \in V(\mathcal{T})$ has a rumor v_r , which may or may not be null and is unknown to all other vertices of \mathcal{T} .

Definition 4.2.2. We allow for vertices to *aggregate* rumors. This means when v receives a rumor from a child \tilde{v} , the rumor $v_r = v_r \cup \tilde{v}_r$.

It is important to note that we will assume that regardless the number of times of aggregation that has occurred, a vertex v may transmit its rumor in one time step. Models without this feature have been considered by [6].

Definition 4.2.3. We say a vertex *transmits* on time step t if a vertex \tilde{v} attempts to send its message to its parent v at time t .

Definition 4.2.4. A *protocol* is a finite collection of instructions for when any subset of the vertices of \mathcal{T} enter the broadcast state. The protocol may depend on the number of vertices, n , the maximum degree of any vertex Δ , the maximum height D , the label of the vertex $v \in [n]$, and any information the vertex has received since the beginning of the protocol until the current time step. The protocol can instruct v to transmit its rumor, or any other information, such as number of children or height.

The importance of this definition is that when considering an arbitrary protocol \mathcal{P} , whatever action it takes at step t may depend on all events up to time $t - 1$. This will be quite important in theorem 7.3.

Definition 4.2.5. A tool that we will make a great deal of use of is called the ALL STEP in which every vertex in the graph enters the broadcast state, this was defined in [6].

Definition 4.2.6. Another useful tool from [6] is the ROUND ROBIN protocol which instructs at time t_j , the vertex with label congruent to $j \bmod n$ will transmit.

Algorithm 1 The ROUND ROBIN Algorithm.

Input: $\{v_1, \dots, v_n\}$, $v_i \in V$

- 1: **for** $i = 1, \dots, \omega$ **do**
 - 2: v_i broadcasts if $t_i \equiv i \pmod n$.
 - 3: **end for**
-

We now need to develop some notation in order to discuss the times in which vertices begin broadcasting which we will define now as their *activation time*.

Definition 4.2.7. For a vertex $v \in \mathcal{T}$, we will call the *activation time* of that vertex the time step in which it begins attempting to transmit to its parent. We will denote it $\alpha(v)$.

In the case of aggregating information, for a vertex $v \in \mathcal{T}$ with children v_1, \dots, v_l , v will not transmit until all children of v have successfully transmitted their rumors to v . We will think of v as waiting for rumors from all of its children so define α as follows:

$$\alpha(v) := \max_{v_i \in v_1, \dots, v_l} \text{time } v_i \text{ succeeds} + 1.$$

In the setting of this thesis, we will focus primarily with investigating the activation time of the root r of the tree \mathcal{T} which is $\alpha(r)$.

Once a vertex becomes active, it will follow a variety of instructions as determined by a protocol \mathcal{P} . Such a protocol will typically have instructions for all n vertices of the tree \mathcal{T} at each time step which vertex will transmit or not.

Remark 4.2.8. In this thesis we will say a protocol succeeds if $\alpha(r)$ is finite for all trees.

Definition 4.2.9. The *length* of a protocol \mathcal{P} is the number of time steps required to complete the protocol, in the case of this thesis it is the maximum value for $\alpha(r)$ over all trees.

Definition 4.2.10. A vertex v *participates* in protocol \mathcal{P} if v transmits according to \mathcal{P} .

Similarly we can discuss when a vertex v no longer participates in a protocol.

Definition 4.2.11. A vertex v *retires* from protocol \mathcal{P} if it has participated in all the steps required by \mathcal{P} .

We now can perform an analysis of algorithm ROUND ROBIN.

Lemma 4.2.12. The algorithm ROUND ROBIN completes in time $\mathcal{O}(n^2)$.

Proof. Let \mathcal{T} be a tree on n vertices. We see that since the instruction given by ROUND ROBIN to vertices is unique there is no chance of collision. We know that every for n steps of the protocol, all vertices \tilde{v} at a fixed height will have successfully broadcast to their parents v . Since the height of the root could be as large as n , it may require n runs of ROUND ROBIN, each having length n achieving a running time of $\mathcal{O}(n^2)$. □

Chapter 5

Tools from Extremal Set Theory

The two most essential constraints of the model are the fact that the protocol \mathcal{P} does not know the topology of the tree \mathcal{T} , and collision. This chapter will present a method first developed by Chrobak, Gasieniec, and Rytter in [7], and with further connections presented by Clementi, Monti, and Silvestri in [8].

In the example 4.1.1, we can imagine a failing protocol as one where the pairs of vertices $\{v_1, v_2\}$ and $\{v_3, v_4\}$ always transmit at the same time, causing collisions to constantly occur and no message ever reaching the root. We wish then to design a protocol to avoid this situation. Furthermore this protocol should function regardless the size of the tree, knowing only that the number of siblings any vertex might have is bounded above by Δ . These “traffic jams” involving $k \leq \Delta$ siblings in the tree will be the motivation for the rest of the work in this chapter.

Let \mathcal{T} be a tree on n vertices and let v be a vertex having k children. We wish to design a protocol so that regardless which labels from n the k children of v are given, in a finite

amount of time, all rumors from the k siblings are successfully transmitted to v . Call this set of siblings W . Clementi, Monti, and Silvestri [8] was the observation that a set theoretic construction of Erdős, Frankl, and Füredi in [11] was precisely what was needed to design such a protocol, using a tool called a strongly selective family. The first use of selectors in general was by Chrobak, Gasenc and Rytter in [7].

5.1 Strongly Selective Families of Sets

While not the original construction of Erdős, Frankl, and Füredi, the version stated is dual to their construction.

Definition 5.1.1. A *strongly k -selective family* is a collection of sets $\mathcal{F} = F_1, \dots, F_m \subset [n]$ satisfying the property that $\forall X \subset [n]$ with $|X| \leq k$, $\forall x \in X, \exists F_i$ where $F_i \cap X = \{x\}$.

This can be translated to a protocol in the following way. The ambient set of size n is the set of all labels of the vertices within \mathcal{T} . At time t , all vertices with labels belonging to F_t will transmit. The strongly k -selective property of the family \mathcal{F} lets us make guarantees about the transmissions of the vertices. For all collections of at most k vertices $W \subset V$, regardless which labels they are given, we can know that by time m , there was a time step i in which $F_i \cap W = \{w\}$. This means that the vertex w successfully transmits to its parent v at time i . Thus the size $|\mathcal{F}|$ is the amount of time required to have a successful transmission from all such collections of siblings to their common parents. Note that we can always obtain a trivial bound by letting F_i be the singleton $\{i\}$ so trivially $|\mathcal{F}| \leq n$. This bound is far from ideal and we will develop tighter bounds now.

However, the question originally asked by Erdős, Frankl, and Füredi had a slightly different statement. This question was asking about the size of families of sets in which no set is covered by the union of r other members of the family. Giving us the original definition from [11]:

Definition 5.1.2. A family \mathcal{G} is a k -cover-free family if $G_0 \not\subset \cup_{i=1}^k G_i$ holds for all distinct sets G_0, \dots, G_r in the family. Let $g_k(m, r)$ denote the maximum cardinality of an k -cover-free family, $\mathcal{G} \subset \binom{X}{r}$ where $|X| = m$.

This statement is dual to the *strongly k -selective family* and can be seen with a bit of translation. Here the G_i can be thought of as the transmission patterns for each vertex where G_i contains the time steps on which i will transmit. The k -union in which no other G_i lies can be thought of as the requirement for a vertex to succeed among k competitors. The ambient set from which G is chosen are the total number of times allowable to broadcast, thus the running time is the size of X , which is m . The value of r is unimportant for our model, but here can be thought of as requiring each vertex transmit exactly r times.

The result of Erdős, Frankl, and Füredi then gives bounds on the behavior of $g_k(m, r)$ as $n \rightarrow \infty$ and $r \ll m$. We then will be looking for a lower bound on $g_k(m, r)$.

Proposition 5.1.3. (Erdős, Frankl, and Füredi) Let $g_k(m, r)$ be as above. Then

$$g_k\left(m, \frac{n}{4k}\right) \geq \left(1 + \frac{1}{4k^2}\right)^m.$$

This statement however, is not quite what we are looking for. While it is not too hard to show that 5.1.3 yields the known upper bound for the size of a strongly k -selective family, we will prove the result in the phrasing of definition 5.1.1, using a probabilistic proof to find an upper bound on the size of the strongly k -selective family \mathcal{F} . This then is

a probabilistic proof of the dual statement of 5.1.3 by Erdős, Frankl, and Füredi which was strictly enumerative when proven in [11]. The statement we are looking for is the following theorem.

Theorem 5.1.4. Let $\mathcal{F} = F_1, \dots, F_m \subset [n]$ be a strongly k -selective family, chosen so that m is minimal. Then

$$|\mathcal{F}| \leq c_1 k^2 \log(n).$$

Proof. We will use the first moment method. Suppose that for every time step, every vertex transmits with probability p and remains silent with probability $1 - p$. Then for a fixed time step, the probability a vertex with k siblings succeeds is

$$\mathbf{P}(v \in \mathcal{T} \text{ succeeds}) = p(1 - p)^{k-1}.$$

So the probability that v succeeds by time t is then

$$\mathbf{P}(v \text{ succeeds by } t) = 1 - (1 - p(1 - p)^{k-1})^t.$$

So, taking expectation over all k -tuples in \mathcal{T} , the event that a vertex fails to successfully transmit to its parent is a failure of the selector property.

$$\mathbf{E}(\text{Failure of the selector property}) = \binom{n}{k} k \left(1 - p(1 - p)^{k-1}\right)^t.$$

Letting $p = \frac{1}{k}$ and using the helpful inequalities,

$$\left(1 - \frac{1}{k}\right)^{k-1} \geq e^{-1},$$

and

$$n^k \geq \binom{n}{k}.$$

This gives us

$$\mathbf{E}(\text{Failure of the selector property}) \leq n^k k \left(1 - \frac{1}{ke}\right)^t \leq n^k k \exp\left(\frac{-t}{ke}\right).$$

And, we can select t to force the expectation go to 0 as $n \rightarrow 0$ as $n \rightarrow \infty$. Letting $t = c_1 k^2 \log(n)$ achieves this. A calculation shows $c_1 > e$ is sufficient. \square

This gives us an important corollary.

Corollary 5.1.5. Let $s_k(n)$ be the minimal size of a strongly k -selective family, which we will call a k -selector. Then for some constant $c_1 > e$,

$$s_k(n) \leq c_1 k^2 \log(n).$$

This finally gives us the bounds we need on the sizes of selectors. The lower bound from Theorem 1.7 in [8] and the upper bound from [11], which we presented an alternate statement and probabilistic proof of for completeness. This gives us the central lemma for selector size.

Lemma 5.1.6. Let $s_k(n)$ be the minimal size of a strongly k -selective family on an ambient set of size n . Then for constants c_1, c_2

$$\frac{c_2 k^2 \log(n)}{\log(k)} < s_k(n) < c_1 k^2 \log(n).$$

Definition 5.1.7. A family of sets $\mathcal{F} = \{F_1, \dots, F_m\}, F_i \subset [n]$ fails to have the *k -selector property* if $\exists X \subset [n], |X| \leq k \exists x \in X, \forall F_i$ where $F_i \cap X \neq \{x\}$.

The purpose of this definition is to understand when a family of sets will not be of use to us in an algorithm. Failing to have the k -selector property will correspond to at least

one child in the tree being unable to ever guarantee successful transmission to its parent. This then leaves us unable to guarantee that the protocol can complete.

Remark 5.1.8. It will be valuable to consider the inverse of $s_k(n) = t$. If we think of $s_k(n)$ as a function that takes the number of vertices we wish to have a k -selective family on, and returns the minimal time to run a k -selective family, $s_k^{-1}(t) = n$ is the maximum number of vertices in which we can have a k -selector run in time t .

It was shown in [8] that should you have too few sets, then you will fail to have the selector property. We will make heavy use of this in proving lower bounds for these algorithms. We can restate the definition 5.1.7 in a way that will be more immediately useful to us. Given without proof is as follows:

Lemma 5.1.9. If \mathcal{F} is a family of sets $\mathcal{F} = \{F_1, \dots, F_m\}$, $F_i \subset [n]$ with $m < \frac{c_2 k^2 \log(n)}{\log(k)}$, then \mathcal{F} fails to have the k -selector property.

Lemma 5.1.10. If a family of sets \mathcal{F} is a strongly k -selective family, then it also is a strongly $(k - i)$ -selective family for all $i \leq k$.

Proof. The definition of the k -selector property $X \subset [n]$, $|X| \leq k \forall x \in X, \exists F_i$ where $F_i \cap X = \{x\}$ and since the size of X can be less than k , trivially X satisfies the $(k - i)$ -selector property. □

In this thesis, we will need to use these families of sets to build algorithms. First we will consider a small example. Let v be a vertex with k children. We will design now an algorithm that regardless of how the children of v are labeled, all their rumors will be successfully transmitted to v .

First we will need to define a function to concretely determine exactly which subsets of the vertex set will attempt to transmit at each time step t . If \mathcal{F} is a strongly k -selective family, each set within $F_t \in \mathcal{F}$ contains the vertices that will broadcast at time t . So our function should map time to these sets. However we will want to follow this procedure more than once, so we will work with $t \equiv i \pmod{s_k(n)}$,

$$\sigma(t, k) : [0, \dots, \omega] \longrightarrow F_i \in \mathcal{F}_k \text{ where } t \equiv i \pmod{s_k(n)}.$$

Algorithm 2 The ONE VERTEX WITH k CHILDREN Algorithm.

Input: v with children v_1, \dots, v_k

Result: v receives rumors from all children.

- 1: All vertices become active
 - 2: **for** $t = 1, \dots, \omega \forall v_i$ child of v **do**
 - 3: **if** $v_i \in \sigma(t, k)$ **and** $t \in [0, S_k)$ **then**
 - 4: v transmits
 - 5: **else**
 - 6: v does not transmit
 - 7: **end if**
 - 8: **end for**
-

Furthermore, we can now build an algorithm that will gather all rumors to the root of a k -regular tree \mathcal{T} with depth D . We will need to make use of the function $\alpha(\cdot)$ now since we cannot start with all vertices active.

Algorithm 3 The k -REGULAR TREE Algorithm.

Input: v with children $v_1, \dots, v_k \in \mathcal{T}$, \mathcal{T} having depth D and root r **Result:** r receives rumors from all vertices in the tree.

```
1: for  $t = 1, \dots, \omega$  do
2:   for  $s = 1, \dots, D$  do
3:     if  $v \in \sigma(t, k)$  and  $t \in [\alpha(v), \alpha(v) + S_k)$  then
4:        $v$  transmits
5:     else
6:        $v$  does not transmit
7:     end if
8:   end for
9: end for
```

We are now in a position to analyze the performance of this algorithm.

Theorem 5.1.11. The algorithm k -REGULAR TREE runs in time

$$\mathcal{O}(Dk^2 \log(n)).$$

Proof. The tree will require S_k steps to advance all rumors from height j to $j + 1$. Thus to advance all rumors a maximum height of D , we will run the k -selector D times yielding the desired bound,

$$\mathcal{O}(Dk^2 \log(n)).$$

□

Chapter 6

Gathering on Degree Three Trees

Before proving the main result, we will give a detailed example to see how these selectors are used in conjunction with knowing the maximum degree of any vertex in the tree \mathcal{T} where we will fix the maximum degree to be 3. These fixed selectors will have sizes we will call S_2 and S_4 . Since the maximum degree in this tree is 3, should enough 4-selectors be run, the protocol will eventually succeed.

6.1 The Degree 3 Tree Algorithm

Algorithm DEGREE 3 TREE will begin by running a 4-selector, so that each vertex may learn how many children it has, and therefore when they can activate. After this, any vertex not receiving a message will know that it is a leaf. These leaves then activate, so we can express this using our terminology by saying $\alpha(v) = 0$. Interior vertices will then wait to hear from all their children before activating. The algorithm will then proceed by rounds, each consisting of three steps: an ALL STEP, a 2-SELECTOR step, and a 4-SELECTOR

step. When a vertex activates, in the first round it is active, it will participate in all three steps. After the first round, it will retire from the ALL STEP and only participate in the 2-SELECTOR and 4-SELECTOR steps of the round. A vertex will participate in 2-SELECTOR steps until it has participated in S_2 steps, which would be at time $\alpha(v) + S_2$ and then no longer participate in 2-SELECTOR steps in future rounds. Similarly it will participate in 4-SELECTOR steps until time $\alpha(v) + S_4$. At this time the vertex will then have retired from the ALL STEP within the round and no longer broadcast, knowing that its information has advanced to its parent. We will now explicitly write this algorithm.

Algorithm 4 The DEGREE 3 TREE Algorithm

Input: $\{v_1, \dots, v_n\}, v_i \in V(\mathcal{T})$

Result: The root r receives all rumors of the vertices of \mathcal{T} .

```

1: Run a 4-SELECTOR
2: for rounds  $t = 1, \dots, \omega, \forall v \in V(\mathcal{T})$  do
3:   for steps  $j = 0, 1, 2$  do
4:     if  $v \in \sigma(t, 2^j)$  and  $t \in [\alpha(v), \alpha(v) + S_{2^j}]$  then
5:        $v$  transmits
6:     else
7:        $v$  does not transmit
8:     end if
9:   end for
10: end for

```

For brevity we will omit the else condition in which v does not transmit in future algorithms.

We will perform our analysis of this algorithm in two parts. We will induct on the height of the tree, supposing that the claims hold for all children of v and then hold for v itself. We then will bound the activation time of an arbitrary vertex $\alpha(v)$ by the function $t(v)$, defined as

$$t(v) = c(h_1(v) + h_2(v)S_2 + h_3(v)S_4).$$

Theorem 6.1.1. If $\alpha(v_i) < t(v_i)$ holds for all children of v then $\alpha(v) < t(v)$ also holds. Then $t(v)$ is an upper bound for $\alpha(v)$.

Recall that

$$\alpha(v) := \max_{v_i \in v_1, \dots, v_l} \text{time } v_i \text{ succeeds} + 1.$$

So we can analyze all scenarios in which children of v interact. Let v_1, v_2, v_3 be children of v . It is possible that v has less than 3 children, however these inequalities will still hold. Let α_1 be the largest time step in which a child of v activates, α_2 largest the second, and α_3 be the third largest activation time. Relabel the children w_1, w_2, w_3 for ease so that $\alpha(w_i) = \alpha_i$. Let \tilde{v} be an arbitrary child of v . We will see that 6.1.3 shows that if $t(\tilde{v}) > \alpha(\tilde{v})$, then $t(v) > \alpha(v)$.

First we will show that

$$\alpha(v) \leq \max\{\alpha_1 + 1, \alpha_2 + S_2 + 1, \alpha_3 + S_4 + S_2 + 1\}.$$

Second we will show that all arguments in the maximum function in 6.1.2 are bounded by $t(v)$ and so,

$$\alpha(v) \leq t(v).$$

Claim 6.1.2. $\alpha(v) \leq \max\{\alpha_1 + 1, \alpha_2 + S_2 + 1, \alpha_3 + S_4 + S_2 + 1\}$.

Proof. Let w_j be a relabeled child of v . Since we know the maximum number of siblings w_j can have, we can go through all potential ways w_j succeeds and has its rumor arrive at its parent v .

1) If both siblings are not transmitting when w_j begins, it will succeed during the ALL STEP, so w_j succeeds on round α_j and by the ordering of the α_i , $\alpha_i < \alpha_1$, we can conclude that for w_j succeeds by round at latest α_1 .

2) If one sibling of w_j activates at the same time, then they will broadcast on the same ALL STEP, so w_j fails on the ALL STEP, but w_j succeeds during the 2-SELECTOR. We can conclude from w_j failing on the ALL STEP that it tells us that there must be a w_k with $\alpha_j = \alpha_k$ so, by the ordering of the α_i , we then know that $\alpha_j \leq \alpha_2$ so w_j succeeds by round $\min\{\alpha_j, \alpha_k\} + S_2 \leq \alpha_2 + S_2$.

3) If both siblings are active when w_j becomes active, and they manage to block each other during the 2-SELECTOR as well as the ALL STEP, but it must succeed during the 4-SELECTOR. The second failure tells us that there must be w_{k_1}, w_{k_2} such that $|\alpha_j - \alpha_{k_i}| < cS_2$ so $\alpha_j \leq \alpha_{k_1} + cS_2$ and $\alpha_j \leq \alpha_{k_2} + cS_2$ so we can conclude $\alpha_j \leq \alpha_3$ so $\min \alpha_j, \alpha_{k_1}, \alpha_{k_2} + cS_4 + cS_2$ so α_j succeeds by round $\alpha_3 + cS_4 + cS_2$.

Since the degree of the tree is at most 3, and every vertex participates in one full four selector guaranteeing all rumors will eventually reach the parent, these cases are exhaustive, and in every instance is bounded by one argument of the maximum of the claim. Thus $\alpha(v) \leq \max\{\alpha_1 + 1, \alpha_2 + S_2 + 1, \alpha_3 + S_4 + S_2 + 1\}$. □

Claim 6.1.3. We will show all arguments of the maximum function of 6.1.2 are bounded by $t(v)$.

Proof. We can break this into three cases.

(1) $t(v) \geq t_1 + 1$ We know from looking at the one heights that $h_1(v) = h_1(w_1) + 1$ so $t(v) - t(w_1) > 1$.

(2) $t(v) \geq t_2 + cS_2$ We know that for two of the children, $i, j \in \{1, 2, 3\}$ of v must have $h_2(v_i) < h_2(v)$ and $h_2(v_j) < h_2(v)$ so $t(v) - t(w_2) > cS_2$ and we then require $cS_2 > S_2 + 1$.

(3) $t(v) \geq t_3 + cS_4$ We know that at least one child $i \in \{1, 2, 3\}$ of v must have $h_3(v_i) < h_3(v)$ so $t(v) - t(w_3) > cS_4$ we then additionally require $cS_4 > S_4 + S_2 + 1$.

□

The choice of $c = \frac{5}{4}$ satisfies all the inequalities, giving us that $\alpha(v) \leq t(v)$.

Theorem 6.1.4. Let \mathcal{T} be a tree with n vertices, each of which having maximum degree 3. Then DEGREE 3 TREE completes in time

$$\mathcal{O}(D + 16 \log(n) \log_3(n)).$$

Proof. We see this by observing the bound on the activation time of the root

$$\alpha(r) \leq \frac{5}{4}(h(r) + h_2(r)S_2 + h_3(r)S_4).$$

Claim 6.1.2 gives us a bound on the activation time for each vertex and 6.1.3 gives the bound stated in the theorem. Thus we can conclude that DEGREE 3 TREE completes in

time

$$\frac{5}{4}(h(r) + h_2(r)S_2 + h_3(r)S_4)$$

as desired. □

Remark 6.1.5. It is important to notice that since the rounds were only comprised of 3 steps, it does not affect the bound as written. However it will appear in the next chapter.

Chapter 7

Gathering on Bounded Degree

Trees

7.1 Degree Δ with Rounds

Our goal now is to show that the success we had on a tree whose maximum degree is three can be extended to any tree with bounded degree. Let the maximum degree of any vertex in \mathcal{T} be $\Delta = \mathcal{O}(\sqrt{n})$.

Lemma 5.1.10 gives the first hint on how to improve the performance of an algorithm. Since all k -selective families are trivially $k-1$ -selective families, we can carry this observation over to algorithms that we run. Then we see that all k -selectors are trivially $k-1$ -selectors, so importantly, not much is gained by running sequential large selectors. So then running fewer selectors with large gaps in their sizes could potentially be one improvement. We then run only 2^i -selectors, which allows us to run only $\log_2(\Delta)$ of the selectors instead of

running all $1, \dots, \Delta$ selectors.

The algorithm will run in rounds, each step of a round will have active vertices participate in one step of each of the $\log_2(\Delta)$ selectors, until retirement.

We will now formally define DEGREE Δ TREE, and then offer a diagram to prepare us for the analysis of the algorithm.

Let $p = \lceil \log_2(\Delta) \rceil$ and let $P = [p] \cup 0$.

Algorithm 5 The DEGREE Δ TREE Algorithm

Input: $\{v_1, \dots, v_n\}$, $v_i \in V(\mathcal{T})$

Result: The root $r \in \mathcal{T}$ receives all rumors of the vertices of \mathcal{T} .

```

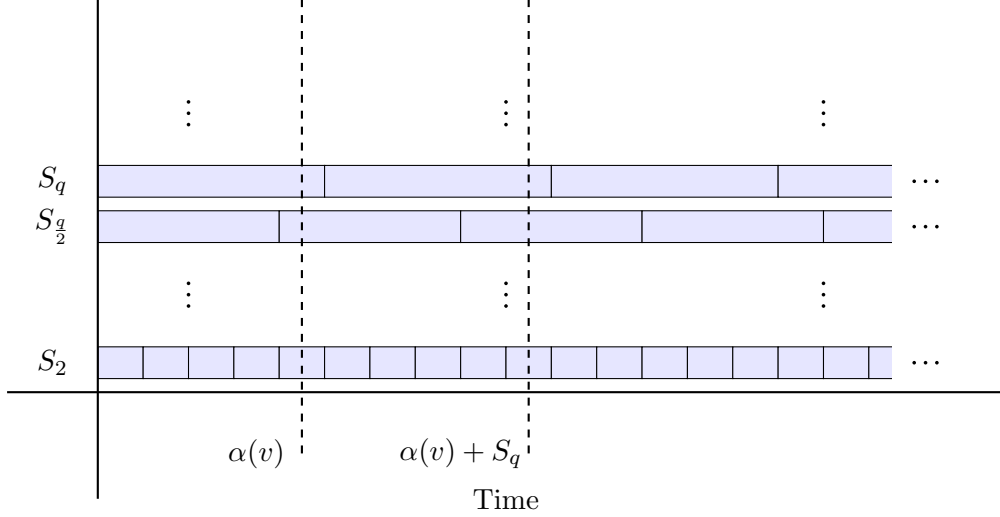
1: Run a  $2^p$ -SELECTOR
2: for rounds  $t = 1, \dots, \omega, \forall v \in V$  do
3:   for steps  $j = 0, \dots, p$  do
4:     if  $v \in \sigma(t, 2^j)$  and  $t \in [\alpha(v), \alpha(v) + S_{2^j}]$  then
5:        $v$  transmits in step  $j$  of round  $t$ 
6:     end if
7:   end for
8: end for

```

Now to help us understand the algorithm, we have the following diagram. One can think of each for loop in the code above being an axis on the following figure. The rounds loop is the time on the x axis, and the steps loop iterates through the various selectors on the y axis.

Thinking of the algorithm from the perspective of any vertex v in the tree we can make

our first important observation. If v becomes active at time $\alpha(v)$, v will have participated in every step of a q selector by round $\alpha(v) + S_q$.



We now need to establish a notion of how vertices may interfere with one another. Consider the following example to see how we might try and classify these collisions.

Example 7.1.1. Let \mathcal{T} be a tree on n vertices with $v, x, y, z \in V$ and let x, y, z be siblings with common parent v . Suppose $\alpha(x) = \alpha(y)$, and that $\alpha(z) = \alpha(x) + S_2 - a$, $0 \leq a \leq S_2$. The vertices x and y block each other on step $\alpha(x)$ and suppose according to the protocol the 2-SELECTOR x would succeed in transmitting its rumor to v on time $\alpha(x) + a$. However, this happens to be the step z became active, and then blocking it during the 2-SELECTOR, thus preventing x from transmitting successfully until participating in a 4-SELECTOR.

This motivates the definition of what we will call collections of vertices that interfere with one another as *in competition*.

Definition 7.1.2. Let \mathcal{T} be a tree on n vertices. Fix an integer $l \leq n$. We will call a vertex v an l -competitor of vertex w if v is a sibling of w and $[\alpha(v), \alpha(v) + S_l] \cap [\alpha(w), \alpha(w) + S_l] \neq \emptyset$.

The motivation here is to understand under what circumstances an l -selector can fail, and determine the window of time in which these events can occur.

Definition 7.1.3. Let \mathcal{T} be a tree on n vertices. We will say a vertex v is in (A, l) -competition with a collection $A \subset V$ if all vertices $w \in A$ are l -competitors of v .

We now can begin to see the strength of these strongly k selective families. As the algorithm progresses for an arbitrary vertex v and its collection of siblings, call them W , we can describe which selectors might be of use to a collection and which would not. Since the degree is bounded and $p = \lceil \Delta \rceil$, there will be a smallest selector S_q , such that v will fail to transmit to its parent on an $S_{\frac{q}{2}}$ step, but will succeed on a S_q step. What we can conclude is that v must have belonged to an (A, j) -competition for $j \in [\frac{q}{2} + 1, q]$ and $A \subset W$, explaining why the q -selector was the first successful selector. Note that v can be in many competitions but this selection of q is unique. From this information we can conclude many things about the structure of the graph local to v . First we know that the parent of v , call it u , must have a $\frac{q}{2} + 1$ -height increase over some child in W , by lemma 2.1.18. So we know that if $h_{\frac{q}{2}+1}(u) - h_{\frac{q}{2}+1}(v) \geq 1$ that a q -selector might be the selector for which v successfully transmits. Thus for every $2^{i-1} + 1$ -height increase, we should attempt to advance the rumor of v with a 2^i selector. This function $t(v)$ does just that and serves as a bound for our running time.

Proposition 7.1.4. Let \mathcal{T} be a tree with bounded degree. The time it takes to gather all information in the subtree rooted at v takes no longer than $t(v)$, where $t(v)$ is defined as

$$t(v) = 2 \left(h_1(v) + \sum_{k=1}^p h_{2^{k-1}+1}(v) S_{2^k} \right).$$

We observe later that 2 may be replaced with $\frac{5}{4}$ but 2 is used for readability.

We have an intermediate functional form of bounding each activation time by the following functions.

$$f_k(v) \text{ where } f_k(v) = \alpha_{2^{k-1}+1} + S_{2^k} + S_{2^{k-1}} + 1 \text{ and } f_0 = \alpha_1 + 1.$$

Theorem 7.1.5. Let \mathcal{T} be a tree with root r on n vertices with maximum degree Δ and depth D . Then DEGREE Δ TREE completes in $\mathcal{O}((D + \Delta^2 \log(n) \log_{\Delta}(n)) \log_2(\Delta))$ steps.

Proof. The time the algorithm completes is $\alpha(r)$, which we will show is bounded by $t(r)$. We will prove this theorem in two parts. First we will bound the activation time $\alpha(v)$ by an intermediate function $f_k(v)$. Then we will show that $f_k(v)$ is bounded by $t(v)$.

We will induct on the height of tree, assuming both claim 7.1.6 and claim 7.1.7 hold for children of v and then showing the claims hold for v itself. The base case is the leaves of the tree, for which both claims hold trivially.

Let v_1, \dots, v_l be the children of v . Relabel these children under $\alpha(\cdot)$, naming them now w_1, \dots, w_l so that $\alpha(w_1) \geq \dots \geq \alpha(w_l)$. We will call this sequence of times $\alpha_1, \dots, \alpha_l$ which is a convenient way of discussing the i^{th} activation time.

With these claims we will be able to prove 7.1.5.

Claim 7.1.6. For all $v \in \mathcal{T}$,

$$\alpha(v) \leq \max_{k \in P} f_k(v).$$

Claim 7.1.7. For all vertices $v \in \mathcal{T}$, $\forall k \in P$ the function $t(v) \geq f_k(v) \forall k \in P$.

Proof. (7.1.6) Let v be a vertex in the tree \mathcal{T} . Let v_1, \dots, v_l be children of v . We know that since $l \leq 2^p$ all rumors will be transmitted to v . Take $\tilde{v} \in v_1, \dots, v_l$ since we know \tilde{v} must succeed, let q the smallest value corresponding to the S_q step of the algorithm for which

\tilde{v} 's rumor is successfully transmitted to v . If $q = 1$, then S_1 , the ALL STEP, is bounded by $\alpha(v) < \alpha_1 + 1$. If $q > 1$, then we know $\frac{q}{2}$ -selector has failed, which means there are at minimum $\frac{q}{2} + 1$ vertices in $\frac{q}{2}$ -competition with \tilde{v} , including \tilde{v} .

Recall for each such vertex w belonging to a collection of siblings with size $\frac{q}{2}$, for the $\frac{q}{2}$ -competitors of \tilde{v} we know that $\alpha(w) \in [\alpha(\tilde{v}) - S_{\frac{q}{2}}, \alpha(\tilde{v}) + S_{\frac{q}{2}}]$. Since we know that there are $\frac{q}{2} + 1$ elements in the interval $[\alpha(\tilde{v}) - S_{\frac{q}{2}}, \alpha(\tilde{v}) + S_{\frac{q}{2}}]$, we know that the $\frac{q}{2} + 1^{\text{th}}$ largest element, which is at least $\alpha_{\frac{q}{2}+1}$. This must be at least as large as $\alpha(\tilde{v}) - S_{\frac{q}{2}}$. Hence $\alpha(\tilde{v}) - S_{\frac{q}{2}} \leq \alpha_{\frac{q}{2}+1}$.

We also have that \tilde{v} successfully transmits its rumor to v by $\alpha(\tilde{v}) + S_q$. Returning to the definition of $\alpha(v)$, v is waiting for the last successful transmission from its child to become active itself, hence

$$\alpha(v) \leq \max_{\tilde{v} \in v_1, \dots, v_l} \text{time } \tilde{v} \text{ succeeds} + 1,$$

and

$$\alpha(\tilde{v}) + S_q \leq \alpha_{\frac{q}{2}+1} + S_q + S_{\frac{q}{2}}.$$

Thus for v we know that $\alpha(v) \leq \alpha_{\frac{q}{2}+1} + S_{\frac{q}{2}} + S_q$ and so for some $k \in P$, $\alpha(v) \leq f_k(v)$ so surely $\alpha(v) \leq \max_{k \in P} f_k(v)$.

□

Proof. (7.1.7) We know that a vertex v with children v_1, \dots, v_l must have γ -height increases over at least some of these children for any value of γ less than l . Relabeling the children of v , v_i according to $\alpha(\cdot)$ as w_i we know from lemma 2.1.18 that for each k there must be

at least one child of v , call it $\tilde{w} \in w_1, \dots, w_{2^{k-1}+1}$ for which $h_{2^{k-1}+1}(v) - h_{2^{k-1}+1}(\tilde{w}) \geq 1$.

Expressing this in term of $t(\cdot)$ we have

$$t(v) - t(\tilde{w}) = 2h_1(v) - 2h_1(\tilde{w}) + 2 \sum_{k=1}^p h_{2^{k-1}+1}(v) S_{2^k} - 2 \sum_{k=1}^p h_{2^{k-1}+1}(\tilde{w}) S_{2^k},$$

grouping terms we obtain

$$2(h_1(v) - h_1(\tilde{w})) + 2 \sum_{k=1}^p (h_{2^{k-1}+1}(v) - h_{2^{k-1}+1}(\tilde{w})) S_{2^k} \geq 2S_{2^k}$$

by the inductive hypothesis of 7.1.5 we have $t(\tilde{w}) \geq \alpha(\tilde{w})$, giving

$$t(v) \geq 2S_{2^k} + \alpha(\tilde{w}).$$

Since \tilde{w} was chosen from a collection of $2^{k-1}+1$ children of v , we know that $\alpha(\tilde{w}) \geq \alpha_{\frac{g}{2}+1}$.

$$t(v) \geq 2S_{2^k} + \alpha_{\frac{g}{2}+1}.$$

Since $2S_k \geq S_{2^k} + S_{2^{k-1}}$ we have found a bound for $t(v)$ on the corresponding f_k . Also note that the $k = 0$ case holds trivially as $2S_k > S_k$. Finally, since for each k , $t(v) \geq f_k(v)$ then

$$t(v) \geq \max_{k \in P} f_k(v)$$

and so $t(v) \geq \alpha(v)$.

□

All rumors then have reached r by time $t(r)$ rounds, so DEGREE Δ TREE completes in time at most

$$t(r) = 2 \left(h_1(r) + \sum_{k=1}^p h_{2^{k-1}+1}(r) S_{2^k} \right) = 2 \left(D + \sum_{k=1}^p h_{2^{k-1}+1}(r) S_{2^k} \right).$$

Making use of the bound 5.1.6 for the selector size, we can arrive at the expression we need for the theorem. Observing the sum is dominated by the last term, and observing that each round is comprised of $\log_2(\Delta)$ steps, we conclude that the performance of DEGREE Δ TREE is

$$\mathcal{O}((D + \Delta^2 \log(n) \log_{\Delta}(n)) \log_2(\Delta))$$

as desired. □

7.2 Degree Δ Without Rounds

In the construction of DEGREE Δ TREE, there were essentially p building blocks that we made use of. First was the 1-SELECTOR or ALL STEP, and the rest were the 2^k -selector for $k \in [1, p]$. We can now think of a way of using these smaller algorithms in a more delicate way than simply running them one after another in the round structure used by DEGREE Δ TREE.

Examining the algorithm DEGREE Δ TREE, it is comprised of rounds that determine a loop of steps to repeat. Each round is made up of $\log_2(\Delta)$ steps, with the first step being the ALL STEP (1-SELECTOR), then a 2-SELECTOR, then a 4-SELECTOR and so on. This essentially gives equal priority to each selector. Another way of looking at the round structure is in the form of guarantees about time between similar steps.

We know that the amount of time between every ALL STEP is $\log_2(\Delta)$, similarly with 2-SELECTOR steps, 4-SELECTOR steps and so on. It might be fruitful then to change that value per step. We could say that we guarantee that the amount of time from one ALL STEP to the next is never greater than β_0 , and the amount of time between any 2-SELECTOR steps is never greater than β_1 and so on. This approach allows us to give different priorities to different components of the protocol. We will now examine this approach, should one exist at all. The existence of such a collection is called the pinwheel problem and is discussed in [12] which observes that a necessary but not sufficient condition is for these gaps to be achievable

$$\sum_{i=0}^p \frac{1}{\beta_i} \leq 1.$$

Finally, we must revisit our definition of σ and add some new functions to help us write the algorithm that makes use of selectors at differing intervals. The first will be ϕ , which will keep track of which selector family to use. The next will be ψ which will keep track of which set within a selector family to run. The last will be σ , which will be the scheduler, using both ϕ and ψ to explicitly say which vertices will transmit on a given time step t . Let ω be the total running time of the algorithm, let $P \subset \mathbb{N}$ be the sizes of the selectors we wish to use and, finally, let $\mathcal{F}_k = \{F_1, \dots, F_m\}$ be a collection of sets satisfying the k -selector property. Then our functions explicitly are:

$$(1) \phi(t) : [0, \omega] \longrightarrow P, t \longmapsto k \text{ where } S_k \text{ is the selector to be used at time } t$$

$$(2) \psi(t, k) : [0, \omega] \times P \longrightarrow \mathbb{N}, (t, k) \longmapsto 1 + |\{\bar{t} : \bar{t} < t, \phi(\bar{t}) = k\}|$$

$$(3) \sigma(t, k) : [0, \omega] \times P \longrightarrow \mathcal{F}, (t, k) \longmapsto F_i \in \mathcal{F}_k \text{ where } \psi(t, k) \equiv i \pmod{S_k}$$

So now for an individual selector, we can explicitly write the algorithm as follows.

Algorithm 6 The k -SELECTOR Algorithm.

Input: $\{v_1, \dots, v_n\}$, $v_i \in V(\mathcal{T})$, $\omega \geq S_k$

Result: Every collection of k siblings advances all rumors to common parent v .

```
1: for  $t = 1, \dots, \omega$ ,  $\forall v \in V(\mathcal{T})$  do  
2:   if  $v \in \sigma(t, k)$  and  $t \in [\alpha(v), \alpha(v) + S_k)$  then  
3:      $v$  transmits  
4:   end if  
5: end for
```

And since it will often be desirable to run many selectors in concert with each other, we can explicitly write how to make a collection P of selectors cooperate. We will use the old framework of rounds in this algorithm in order to see how we would make use of these new helper functions to write the algorithm explicitly.

Algorithm 7 The MULTIPLE SELECTOR Algorithm.

Input: $\{v_1, \dots, v_n\}$, $v_i \in V(\mathcal{T})$, Collection of selectors P , $\omega \geq \max\{S_{k_1}, \dots, S_{k_{|P|}}\}$.**Result:** Every collection of siblings that are smaller than the largest value in P advance all rumors to common parent v .

```
1: for  $t = 1, \dots, \omega$ ,  $\forall v \in V(\mathcal{T})$  do
2:   for  $j = 1, \dots, |P|$  do
3:     if  $v \in \sigma(t, \phi(t))$  and  $t \in [\alpha(v), \alpha(v) + S_{\phi(t)}]$  then
4:        $v$  transmits
5:     end if
6:   end for
7: end for
```

This lets us concretely write an algorithm for these requirements. Let ϕ, ψ, σ be chosen for a sufficient collection of $\{\beta_i\}_{i \in I}$. Then critically we can write the algorithm now without rounds. Notice that only one loop is required in this algorithm.

Algorithm 8 The β -WEIGHTED STEPS Algorithm

```
1: Run a  $2^P$ -selector
2: for  $t = 1, \dots, \omega$ ,  $\forall v \in V$  do
3:   if  $v \in \sigma(t, \phi(t))$  and  $t \in [\alpha(v), \alpha(v) + \beta_{\phi(t)} S_{\phi(t)}]$  then
4:      $v$  transmits
5:   end if
6: end for
```

Assuming that a suitable collection of $\{\beta_i\}_{i \in I}$ can be found, our function $t(v)$ and intermediate bounds for $\alpha(v)$ can be updated to accommodate this strategy. We construct a new function to bound the activation time of any vertex as follows,

$$t'(v) = c_0 h_1(v) + \sum_{k=1}^p c_k h_{2^{k-1}+1}(v) S_{2^k}.$$

Noticing that the constant of 2 in claim 7.1.5 has been replaced by the yet to be determined constant c_k . We can also adapt the activation times of any vertex to incorporate these betas as well, modifying the observation on activation times made in DEGREE Δ TREE, now observing that a vertex v will retire from a q -selector at time $\alpha(v) + \beta_q S_q$ so we obtain

$$f'_k(v) \text{ where } f'_k(v) = \alpha_{2^{k-1}+1} + \beta_{2^k} S_{2^k} + \beta_{2^{k-1}} S_{2^{k-1}} \text{ and } f'_0 = \alpha_1 + 1.$$

Since we are working in steps and not rounds, we need to modify the definition of competitor to account for the choice of the collection $\{\beta_i\}$.

Definition 7.2.1. Let \mathcal{T} be a tree on n vertices. Fix an integer $l \leq \Delta$. We will say a vertex v is an (β, l) -competitor of vertex w if v is a sibling of w and $[\alpha(v), \alpha(v) + \beta S_l] \cap [\alpha(w), \alpha(w) + \beta S_l] \neq \emptyset$.

Definition 7.2.2. Let \mathcal{T} be a tree on n vertices. We will say a vertex v is in (β, A, l) -competition with a collection $A \subset V$ if all vertices $w \in A$ are l -competitors of v .

Theorem 7.2.3. Let \mathcal{T} be a tree on n vertices with any labeling having maximum depth D and maximum degree Δ . Let $\{\beta_i\}_{i \in I}$ be associated with the S_{2^i} steps of the algorithm. Then algorithm β -WEIGHTED STEPS completes in

$$\mathcal{O}(\beta_0 D + \sum_{k=1}^p (\beta_{2^k} + \beta_{2^{k-1}}) S_{2^k} \log_{2^{k-1}+1}(n)).$$

Proof. We prove this in two parts. Similarly to 7.1.5 we will induct on the height of an arbitrary vertex v and use the following two claims.

Claim 7.2.4. Let \mathcal{T} be a tree with root r , having maximum degree Δ then for all $v \in \mathcal{T}$,

$$\alpha(v) \leq \max_{k \in P} f'_k(v)$$

and

Claim 7.2.5. For a tree \mathcal{T} with bounded degree, $\forall v \in \mathcal{T}, \forall k \in P$, the function $t'(v) \geq f'_k(v) \forall k \in P$.

Proof. (7.2.4) This follows the proof of 7.1.6 closely. Let v be a vertex in the tree \mathcal{T} . Let $W, |W| = l$ be children of v . We know that since $l \leq 2^p$, all rumors will be transmitted to v . Take $\tilde{v} \in W$, since we know \tilde{v} must succeed, let q be the smallest value corresponding to the S_q step of the algorithm for which \tilde{v} 's rumor is successfully transmitted to v , and for which v failed during the $\frac{q}{2}$ selector. If $q = 1$, then S_1 , the ALL STEP, is bounded by $\alpha(v) < \alpha_1 + 1$. If $q > 1$, then we know $\frac{q}{2}$ -selector has failed, which means there are at minimum $\frac{q}{2} + 1$ vertices in $(\beta_{\frac{q}{2}}, \frac{q}{2})$ -competition with \tilde{v} , including \tilde{v} .

Recall for each such vertex w belonging to a collection of siblings with size $\frac{q}{2}$, for the $(\beta_{\frac{q}{2}}, \frac{q}{2})$ -competitors of \tilde{v} we know the following: $\alpha(w) \in [\alpha(\tilde{v}) - \beta_{\frac{q}{2}} S_{\frac{q}{2}}, \alpha(\tilde{v}) + \beta_{\frac{q}{2}} S_{\frac{q}{2}}]$. Since

we know that there are $\frac{q}{2} + 1$ elements in the interval $[\alpha(\tilde{v}) - \beta_{\frac{q}{2}}S_{\frac{q}{2}}, \alpha(\tilde{v}) + \beta_{\frac{q}{2}}S_{\frac{q}{2}}]$, and we know that the $(\frac{q}{2} + 1)^{th}$ largest element, which is at least $\alpha_{\frac{q}{2}+1}$. This must be at least as large as $\alpha(\tilde{v}) - \beta_{\frac{q}{2}}S_{\frac{q}{2}}$. Hence $\alpha(\tilde{v}) - \beta_{\frac{q}{2}}S_{\frac{q}{2}} \leq \alpha_{\frac{q}{2}+1}$.

We also have that \tilde{v} successfully transmits its rumor to v by $\alpha(\tilde{v}) + \beta_q S_q$. Returning to the definition of $\alpha(v)$, v is waiting for the last successful transmission from its child to become active itself, hence

$$\alpha(v) \leq \max_{\tilde{v} \in v_1, \dots, v_l} \text{time } \tilde{v} \text{ succeeds} + 1$$

and

$$\alpha(\tilde{v}) + \beta_q S_q \leq \alpha_{\frac{q}{2}+1} + \beta_q S_q + \beta_{\frac{q}{2}} S_{\frac{q}{2}}.$$

Thus for v we know that $\alpha(v) \leq \alpha_{\frac{q}{2}+1} + \beta_{\frac{q}{2}} S_{\frac{q}{2}} + \beta_q S_q$ and so for some $k \in P$, $\alpha(v) \leq f'_k(v)$ so surely $\alpha(v) \leq \max_{k \in P} f'_k(v)$.

□

Proof. (7.2.5) This follows the proof of 7.1.7 closely. We know that a vertex v with children v_1, \dots, v_l must have γ -height increases over at least some of these children for any value of γ less than l . Relabeling the children of v , v_i according to $\alpha(\cdot)$ as w_i we know from lemma 2.1.18 that for each k there must be at least one child of v , call it $\tilde{w} \in w_1, \dots, w_{2^{k-1}+1}$ for which $h_{2^{k-1}+1}(v) - h_{2^{k-1}+1}(\tilde{w}) \geq 1$. Expressing this in term of $t'(\cdot)$ we have

$$t'(v) - t'(\tilde{w}) = \beta_0(h_1(v) - h_1(\tilde{w})) + \sum_{k=1}^p c_k h_{2^{k-1}+1}(v) S_{2^k} - \sum_{k=1}^p c_k h_{2^{k-1}+1}(\tilde{w}) S_{2^k},$$

grouping terms we obtain

$$\sum_{k=1}^p (h_{2^{k-1}+1}(v) - h_{2^{k-1}+1}(\tilde{v})) c_k S_{2^k} \geq c_k S_{2^k}$$

by the inductive hypothesis of 7.2.3 we have $t(\tilde{w}) \geq \alpha(\tilde{w})$, giving

$$t'(v) \geq c_k S_{2^k} + \alpha(\tilde{w}).$$

Since \tilde{w} was chosen from a collection of $2^{k-1}+1$ children of v we know that $\alpha(\tilde{w}) \geq \alpha_{\frac{q}{2}+1}$.

$$t'(v) \geq c_k S_{2^k} + \alpha_{\frac{q}{2}+1}.$$

We require $c_k S_k \geq \beta_{2^k} S_{2^k} + \beta_{2^{k-1}} S_{2^{k-1}}$, and we have found a bound for $t'(v)$ on the corresponding f'_k . Furthermore note that this holds trivially for the $k = 0$ case. Finally, since for each k , $t'(v) \geq f'_k(v)$ then

$$t'(v) \geq \max_{k \in P} f'_k(v)$$

and so $t'(v) \geq \alpha(v)$. □

This then lets us examine the time for which the root activates, which is the running time of the algorithm β -WEIGHTED STEPS. We have shown so far that $t'(r) \geq \alpha(r)$. Examining $t'(r)$, we can use the upper bound on the selector size yet again, and we have that $h_1(r) = D$. Finally we have from our requirements on the constants c_k we have that $t'(r) = \mathcal{O}(c_0 D + \sum_{k=1}^p c_k S_{2^k} \log(n) \log_{2^{k-1}+1}(n))$. Which back in terms of the $\{\beta_i\}$ is

$$\mathcal{O} \left(\beta_0 D + \sum_{k=1}^p (\beta_{2^k} + \beta_{2^{k-1}}) S_{2^k}^2 \log(n) \log_{2^{k-1}+1}(n) \right).$$

□

This all supposed that an amenable collection $\{\beta_i\}_{i \in I}$ was chosen. We will see shortly that picking the β_i to be consecutive powers of 2 will indeed give us an algorithm, moreover we suspect this algorithm is optimal.

To write the algorithm explicitly, we will need to define a small function to help us determine what instruction to follow at time t . Define $b(t)$ in the following way:

$$b(t) = \min_j t \not\equiv 0 \pmod{2^j}$$

which means for this algorithm we can explicitly state $\phi(t)$, for t even,

$$\phi(t) = 2^{p-b(t)+2}.$$

Concretely, the algorithm now can be stated as follows:

Algorithm 9 The 2^i -WEIGHTED STEPS Algorithm

- 1: Run a 2^p -SELECTOR
 - 2: **for** $t = 1, \dots, \omega, \forall v \in V$ **do**
 - 3: **if** i odd and $i \in [\alpha(v), \alpha(v) + \beta_0 S_1]$ **then**
 - 4: v participates in an all step
 - 5: **end if**
 - 6: **if** t even and $v \in \sigma(t, \phi(t))$ and $t \in [\alpha(v), \alpha(v) + \beta_{\phi(t)} S_{\phi(t)}]$ **then**
 - 7: v transmits
 - 8: **end if**
 - 9: **end for**
-

Theorem 7.2.6. Let \mathcal{T} be a tree on n vertices with maximum degree Δ and depth D . Let $\{\beta_{2^i}\}_{i \in P}$ be consecutive powers of 2. Then the algorithm 2^i -WEIGHTED STEPS exists and finishes in finite time.

Proof. The existence of the protocol depends on having a unique instruction per step. Since the values for β_i were chosen to be powers of 2, the smallest j for which the residue of step $t \bmod 2^j$ is nonzero means on step t , vertices participate in the selector associated with step β_j . Since this value of j is unique, there is no ambiguity for what instruction to run on step t . Finally, we see that this algorithm will in fact complete, since $2^p > \Delta$ then the maximum number of steps required will be $\beta_{2^p} S_{2^p} D$. \square

We will see that a more careful assignment of the β_i can improve performance significantly. Furthermore, taking $c_k = (2^{p-k+2} + \frac{1}{4})$, we are able to return to 7.2.3 and compute $\mathcal{O}(t'(r))$ exactly.

Theorem 7.2.7. Let \mathcal{T} be a tree on n vertices with any labeling, having maximum depth D and maximum degree Δ . Let $\{\beta_i\}_{i=1}^p = 2^{p-i+2}$ and $\beta_0 = 2$. Then

$$\mathcal{O}(t'(r)) = \mathcal{O}(D + \Delta^2 \log(n) \log_{\Delta}(n)).$$

Proof. We simply replace all values from our earlier result for $t'(v)$ with r and constraints of the tree, and observe $h_1(r) = D$ and using the bound obtained on the selector size in lemma 5.1.6 we obtain the bound found in 7.2.3. First we can replace the value

$$t'(v) = c_0 D + \sum_{k=1}^p c_k S_{2^k} \log_{2^{k-1}+1}(n)$$

to get

$$t'(v) = 2D + \sum_{k=1}^p 2^{p-k+2} S_{2^k} \log_{2^{k-1}+1}(n).$$

We now require the $k = p$ term to dominate to complete the theorem. Expanding the sum and replacing S_{2^k} with the upper bound obtained in 5.1.6, albeit omitting the constant

for clarity since we are concerned with asymptotic behavior, we see the following:

$$\left((2^{p+1} + \frac{1}{4})\right)2^2 \log_2(n) \log(n) + \left(2^p + \frac{1}{4}\right)4^2 \log_3(n) \log(n) + \left(2^{p-1} + \frac{1}{4}\right)8^2 \log_5(n) \log(n) + \dots$$

So we see that we have a function exponential in k and a log whose base is increasing in k , so for the asymptotic behavior, we will be looking at the interaction between 2^{p-k+2} and S_{2^k} . Starting by changing the base of the logarithm that depend on k and reorganizing the sum we have

$$c_1 2^p \log^2(n) \sum_{k=1}^p 2^k \frac{1}{\log(2^{k-1} + 1)}.$$

Again observing that we are dropping the constant from $s_k(n)$ that would be present for clarity. Splitting the sum at $\lfloor \frac{p}{2} \rfloor$ we have

$$\sum_{k=1}^{\lfloor \frac{p}{2} \rfloor} 2^k \frac{1}{\log(2^{k-1} + 1)} + \sum_{k=\lfloor \frac{p}{2} \rfloor + 1}^p 2^k \frac{1}{\log(2^{k-1} + 1)}.$$

Seeing that the first sum is dominated by $2^{\lfloor \frac{p}{2} \rfloor + 1} - 1$ and we can replace the second sum by taking $\frac{1}{\log(2^{k-1} + 1)}$ out, obtaining:

$$2^p \log^2(n) \sum_{k=1}^p 2^k \frac{1}{\log(2^{k-1} + 1)} \leq 2^{\lfloor \frac{p}{2} \rfloor + 1} - 1 + \frac{1}{\log(2^{k-1} + 1)} + \sum_{\lfloor \frac{p}{2} \rfloor + 1}^p 2^k.$$

And since

$$2^p \geq \sum_{k=1}^{p-1} 2^k,$$

we obtain

$$\mathcal{O} \left(2^{\lfloor \frac{p}{2} \rfloor + 1} - 1 + \frac{1}{\log(2^{k-1} + 1)} \sum_{\lfloor \frac{p}{2} \rfloor + 1}^p 2^k \right) = \mathcal{O} \left(\frac{2^p}{\log(2^p)} \right).$$

Since $2^p = \mathcal{O}(\Delta)$ we can now write the asymptotic bound of $t'(v)$

$$t'(r) = h_1(r) + \sum_{k=1}^p \left(2^{p-k+2} + \frac{1}{4} \right) h_{2^{k-1}+1}(r) S_{2^k} = \mathcal{O}(D + \Delta^2 \log(n) \log_{\Delta}(n))$$

as desired. □

7.3 Lower Bounds for Complete Tree

Recall definition 4.2.4, where the framework for an arbitrary protocol was established.

We now wish to consider specifically a protocol designed to deliver all rumors in a tree to the root.

Let \mathcal{P} be a protocol that aggregates all rumors on a tree. Let \mathcal{T} be a regular k -ary tree with n leaves to the root r . The only data \mathcal{P} has to begin with is n and k and \mathcal{P} must aggregate all rumors in \mathcal{T} to r .

From this data, the instruction to transmit or remain silent that \mathcal{P} gives to a vertex v at time t can depend on the following:

- (1) The label of v and the total number of vertices n .
- (2) The current time step t .
- (3) The labels of all vertices v has received a rumor from the start of the protocol until the current time t .
- (4) The contents of the rumor and any additional information a child transmits to v from the start of the protocol until the current time t .

The instruction \mathcal{P} then gives is to transmit or remain silent. There are two possible outcomes for any vertex v that \mathcal{P} instructs to transmit. Either v transmits its rumor or other information dictated by \mathcal{P} to its parent successfully, and we will call this a success, or v transmits but collides with a sibling transmitting on the same step and we will call this a failure.

If we were to attempt to delay the success of an arbitrary protocol \mathcal{P} , we would attempt to maximize the following expression

$$\min_{\text{protocols}} \max_{\text{trees}} \text{running time.}$$

In our case, we will restrict the discussion to complete k -ary trees. This changes the statement to maximizing the expression:

$$\min_{\text{protocols}} \max_{\substack{\text{trees} \\ \text{depth} = D \\ \text{degree} = k}} \text{running time.}$$

We now wish to assign labels to the tree \mathcal{T} in a way to maximally delay \mathcal{P} .

We now have a general framework to introduce some notation that will let us keep track of how long we can delay a given vertex in the tree.

We will use a result in [8], showing that there exists a $c_2 > 0$ such that

$$m \leq \frac{c_2 k^2 \log(n)}{\log(k)}$$

so then 5.1.6 tells us that should we have a family of sets with only m elements, the m sets will fail to have the k -selector property.

As stated in chapter 5, we know that if $s_k(n)$ is the minimal size of the family \mathcal{F} satisfying the k -selector property, then in terms of an algorithm we can say $|\mathcal{F}|$ is the time required to run a k -selector.

There is also a way of looking at this from the perspective of the inverse. Given a fixed amount of time t , let $s_k^{-1}(t)$ be the largest number of vertices where t is sufficient time to run a k -selector, or the largest size of $|\mathcal{F}|$ where \mathcal{F} still satisfies the k -selective property on an ambient set with size at most $s_k^{-1}(t)$.

Fix a length of time in steps, call it \hat{t} so that $\hat{t} = s_k(n^{c_3})$ for some constant $c_3 < 1$ and let $s_k(n) = t$.

The structure of the proof of the lower bound will go as follows. Starting with the leaves, condition the protocol \mathcal{P} on the fact we know the transmission patterns of the leaves from time 0 to time \hat{t} will be determined by the fact that the leaves receive no information. This will give a concrete sequence of transmission patterns for each leaf F_1, \dots, F_n , each F_i dictating which time steps between 0 and \hat{t} vertex i will attempt to transmit. Then we will group these transmission patterns in k -tuples, so that each k -tuple fails to have the k -selector property. We can do this as many times as possible until the remaining tuples satisfy the k -selector property. Thus, if there are n vertices there are $\frac{(k-1)n}{k}$ leaves, we can form

$$\frac{\frac{(k-1)n}{k} - s_k^{-1}(\hat{t})}{k}$$

k -tuples in this way.

So for $\hat{t} = s_k(n^{c_3})$ with $c_3 < 1$, we know that there exists a relabeling of the vertices of \mathcal{T} so that we may delay the advancement of messages from many groups of k siblings for \hat{t} steps.

We now have an idea about how to proceed. Every \hat{t} steps we proceed, we will turn our attention to vertices one height higher in the tree conditioning on the behavior of the vertices we have grouped. So from time 0 to time $(j+1)\hat{t}$, we will look at the behavior of the vertices at height $j-1$ which have been ground and then tell us the behavior of vertices at height j . This then will give us the transmission patterns of the vertices at height j from time $j\hat{t}$ to time $(j+1)\hat{t}$. We then can examine these transmission patterns and group them

into k -tuples which fail to have the selector property, hence fail to transmit to vertices at height $j + 1$.

So given a k -ary tree, we can now state our lower bound as follows.

Theorem 7.3.1. Let \mathcal{P} be an arbitrary protocol as above. Let \mathcal{T} be a regular k -ary tree having n leaves and root r . Then for $c_3 < 1$ there is a labeling of the vertices of \mathcal{T} where

$$\alpha(r) > \frac{(1 - c_3)c_2k^2 \log(n^{c_3}) \log(n)}{\log^2(k)},$$

and so \mathcal{P} runs in time

$$\Omega\left(\frac{k^2 \log(n^{c_3}) \log(n)}{\log^2(k)}\right).$$

Proof. Let \hat{t} be chosen as above. Let \mathcal{P} be an arbitrary protocol on \mathcal{T} . Thinking of \mathcal{T} as a complete unlabeled k -ary tree, we will assign labels from the leaves upward in a way to maximize $\alpha(r)$. Let v be a vertex in the tree at height j that has still not yet received messages from its children by $j\hat{t}$, and let the number of such vertices where this is the case be q_j . Then q_1 is the number of parents at height 1 still waiting by time \hat{t} to hear from their k children. Recall that from the leaves, we were able to see

$$q_1 = \frac{n - s_k^{-1}(\hat{t})}{k}.$$

We now can form a recursion on q_j , by following the procedure outlined above. The number of vertices waiting at time $j\hat{t}$ is

$$q_j = \frac{q_{j-1} - s_k^{-1}(\hat{t})}{k}.$$

We will bound below the quantity q_j by a_j , which has yet to be determined. Let $q_0 = a_0 = n$ be the number of leaves.

Lemma 7.3.2. If $a_0 = n$, then the recursion

$$a_j = \frac{1}{k} (a_{j-1} - n^{c_3})$$

is positive so long as $j > (1 - c_3) \log_k(n)$.

Proof. Solving the recursion gives

$$a_j = \frac{k^{2-j}n - k^{1-j}n - n^{c_3} + n^{c_3}k^{1-j}}{k - 1}$$

observing that a_j is positive until $j = (1 - c_3) \log_k(n)$. □

We now will now use this in terms of our protocol, starting again with the leaves. Condition \mathcal{P} on the leaves of \mathcal{T} not receiving any rumor until time \hat{t} . Then examining transmission patterns of all the leaves as dictated by \mathcal{P} conditioned all events until \hat{t} , group the leaves into k -tuples, corresponding to k -tuples of transmission patterns that fail to have the k -selector property. We can form $\frac{n - s_k^{-1}(\hat{t})}{k}$ k -tuples and assign these tuples parents. These parents will not be able to activate by time \hat{t} in this construction.

So for vertices at height j , q_j of which still have yet to activate, we wish to group them into k tuples that fail to have the selector property. To do this, we need to understand the times in which these vertices will attempt to transmit. This means we need to condition the protocol on the behavior of the child vertices at height $j - 1$, whose transmission pattern will determine the firing pattern of vertices at height j . We then will group these vertices into k tuples so that their transmission patterns fail to have the k -selector property from time $j\hat{t}$ to $(j + 1)\hat{t}$ and assign them parents. As observed earlier, we can form

$$\frac{q_j - s_k^{-1}(\hat{t})}{k}$$

tuples in this way.

So from our original recursion, replacing q_j by its lower bound a_j , we have

$$a_j \geq \frac{1}{k} (a_{j-1} - n^{c_3}).$$

So for the protocol, so long as $q_j > 0$, we know that \mathcal{P} has not completed. So, the quantity we can control is a_j and we will then see how long (meaning for as large a j as possible) $a_j > 0$, which is what the lemma 7.3.2 gave us.

This means we can guarantee the protocol has not finished until time

$$(1 - c_3) \log_k(n) \hat{t} = \frac{(1 - c_3) c_2 k^2 \log(n^{c_3}) \log(n)}{\log^2(k)}.$$

So we obtain a lower bound for the running time of \mathcal{P} ,

$$\Omega\left(\frac{k^2 \log(n^{c_3}) \log(n)}{\log^2(k)}\right)$$

as desired. □

Chapter 8

On the Number of Integral Graphs

Recall that a graph is called integral if the eigenvalues of its adjacency matrix are integers.

First discussed in chapter 3, we now wish to consider how common these graphs are.

Let I_n be the number of integral graphs on n vertices. We now ask what portion of all graphs on n vertices are integral. In a probabilistic language, if G is any graph on n vertices we wish to understand

$$\mathbf{P}(G \text{ is integral}) = \frac{I_n}{2^{\binom{n}{2}}}$$

as n tends to infinity.

Ahmadi, Alon, Blake, and Shparlinski gave the first non-trivial bound in 2009 which stated in a probabilistic language is

Theorem 8.0.1. The probability that a randomly chosen graph on n vertices is integral is, for sufficiently large n , at most $2^{-n/400}$.

They noted in their paper that “we believe our bound is far from being tight and the number of integral graphs is substantially smaller”. Our main result confirms this belief, showing that the proportion of integral graphs decays much faster than exponentially.

Theorem 8.0.2. The probability that a randomly chosen graph on n vertices is integral is, for large n , at most $2^{-cn^{3/2}}$, for some absolute constant c .

Remark 8.0.3. This result is likely still not close to being tight. For more on this, see the final section of this chapter.

8.1 Deterministic properties of adjacency matrices

Now we will collect a few linear algebraic properties that hold for the adjacency matrix of every graph, deterministic or random. The main result of this chapter will follow from combining these properties with a counting argument, essentially tracking how the spectrum of the adjacency matrix behaves as the graph grows.

We begin with the quick observation that adjacency matrices of graphs cannot have too many large eigenvalues.

Lemma 8.1.1. Let G be an arbitrary graph on n vertices, and let A be the adjacency matrix of G . Then A must contain at least $\frac{3n}{4}$ eigenvalues in the interval $[-2\sqrt{n}, 2\sqrt{n}]$.

Remark 8.1.2. Up to the constant of 2, this bound is tight, as it follows from Wigner’s semicircular law 3.1.1 together with interlacing that for large n almost every graph on n vertices has at least $n/3$ eigenvalues outside the interval $[-\sqrt{n}, \sqrt{n}]$.

Proof. Let λ_i be the i^{th} eigenvalue of A . We have

$$\sum_{i=1}^n \lambda_i^2 = \text{Tr}(A^2) = 2E(G) \leq n^2,$$

where $E(G)$ is the number of edges of G . Let T be the multiset of eigenvalues of A that are at least $2\sqrt{n}$ in absolute value. Then we also have

$$\sum_{i=1}^n \lambda_i^2 \geq (2\sqrt{n})^2 |T|.$$

Combining the above two bounds, we have $|T| \leq \frac{n}{4}$ so at least $\frac{3n}{4}$ eigenvalues lie inside the interval.

□

In particular, this implies that any integral matrix must have a large amount of multiplicity in its spectrum: The average multiplicity of an integer in $[-2\sqrt{n}, 2\sqrt{n}]$ as an eigenvalue is proportional to \sqrt{n} . Tao and Vu [19] have shown that having even a single eigenvalue repeated even twice is (polynomially) unlikely for a random matrix, and the goal will be to show that such a large amount of repetition is far less likely. For this our perspective, drawing on an idea originally due to Komlós [15], will be to think of A as being grown “minor by minor” (equivalently, we will expose the graph G vertex by vertex).

Let A_k be the upper left $k \times k$ minor of A . Then A_{k+1} has the block structure

$$A_{k+1} = \begin{pmatrix} A_k & x_{k+1} \\ x_{k+1}^T & 0 \end{pmatrix}$$

where x_{k+1} is the newly added column. Our second observation (a variation on Lemma 2.1 from [19]) is that whenever an eigenvalue’s multiplicity increases from A_k to A_{k+1} , the new column must satisfy certain orthogonality conditions.

Lemma 8.1.3. Let A_k , A_{k+1} , and x_{k+1} be as above. Let λ be any eigenvalue whose multiplicity in the spectrum of A_{k+1} is strictly larger than in A_k , and let v be any eigenvector of A_k corresponding to λ . Then x_{k+1} and v are orthogonal.

Proof. Let w be an arbitrary eigenvector of A_{k+1} , and write $w = \begin{pmatrix} w' \\ w^{(k+1)} \end{pmatrix}$, where $w' \in \mathbb{R}^k$ and $w^{(k+1)} \in \mathbb{R}$. Note that if $w^{(k+1)} = 0$, then w' is itself an eigenvector of A_k with the same eigenvalue. In particular, since the multiplicity of λ increases from A_k to A_{k+1} , there must be an eigenvector w with $Aw = \lambda w$ and $w^{(k+1)} \neq 0$. Fix such a w .

Now let v_1, \dots, v_k be an orthonormal eigenbasis for A_k . After changing our basis for \mathbb{R}^{k+1} to $\{v_1, \dots, v_k, e_{k+1}\}$ (where e_{k+1} is the standard basis vector), we may without loss of generality assume that A_k is diagonal, with the eigenvalues corresponding to λ appearing in the first j coordinates for some j . For any $1 \leq i \leq j$, the i^{th} coordinate of the eigenvalue equation $A_{k+1}w = \lambda w$ now gives

$$\lambda w^{(i)} = \lambda w^{(i)} + w^{(k+1)} x_k^{(i)}$$

where $x_k^{(i)}$ denotes the i^{th} coordinate of x_k . Since by assumption $w^{(k+1)} \neq 0$, we must have $x_k^{(i)} = 0$. Returning to the original basis, this corresponds to x_k being orthogonal to v_i .

This is true for every i between 1 and j , so x_k is orthogonal to the entire eigenspace. \square

Finally, we will make use of the following observation of Odlyzko [18], whose proof we include here for completeness.

Lemma 8.1.4. Let S be an arbitrary $n - \ell$ dimensional subspace of \mathbb{R}^n . Then S contains at most $2^{n-\ell}$ vectors from $(0, 1)^n$. Equivalently, S contains at most a $2^{-\ell}$ proportion of all $(0, 1)$ vectors.

Proof. Since S has dimension $n - \ell$, there must be a collection of $n - \ell$ coordinates which parameterize the space, in the sense that those coordinates uniquely determine the remaining ℓ coordinates. There are $2^{n-\ell}$ choices for the values of those coordinates. \square

8.2 The Proof of Theorem 8.0.2

The rough idea of our argument will be to track the growth of the each integer's multiplicity as an eigenvalue as k increases. On the one hand, we know from Lemma 8.1.1 that by the end of the process there must be a large amount of total multiplicity, in the sense that the average multiplicity of an integer in $[-2\sqrt{n}, 2\sqrt{n}]$ as an eigenvalue is large. This will imply there must be many distinct pairs (k, λ) where some eigenvalue λ that already has large multiplicity increases its multiplicity further as A_k is augmented to A_{k+1} . Now Lemma 8.1.3 will imply we have many orthogonality relations, which will turn out to be unlikely for a random matrix. We now turn to the details.

Given an $n \times n$ matrix A and an integer i between $-2\sqrt{n}$ and $2\sqrt{n}$, recall that $\text{Mult}(A_k, i)$ is the multiplicity of i as an eigenvalue of A_k , and let $\text{Mult}(A, i) = \text{Mult}(A_n, i)$ be its multiplicity in the spectrum of A . By Cauchy Interlacing (e.g. [13] Theorem 4.3.8) we have

$$|\text{Mult}(A_k, i) - \text{Mult}(A_{k+1}, i)| \leq 1.$$

In particular, i must attain each multiplicity between 0 and $\text{Mult}(A, i)$ at least once during the augmentation process. With this in mind, we define the vector a_i as follows:

$$a_i(m) = \begin{cases} 0, & \text{if } \text{Mult}(A_m, i) > \text{Mult}(A, i) \\ 0, & \text{if there is some } m' > m \text{ with } \text{Mult}(A_{m'}, i) = \text{Mult}(A_m, i) \\ j, & \text{if } m \text{ is the largest } k \text{ with } \text{Mult}(A_k, i) = j \text{ and } j \leq \text{Mult}(A, i) \end{cases}$$

The role of the vectors a_i here is to track how the multiplicity of each eigenvalue of A increases throughout the augmentation process. We make the following observations about the vectors a_i :

- For each $m < n$ with $a_i(m) \neq 0$, the multiplicity of i as an eigenvalue must necessarily increase as A_m is augmented to A_{m+1} , since otherwise m would not be the largest minor with this multiplicity. This is clearly the case if $\text{Mult}(A_{m+1}, i) = \text{Mult}(A_m, i)$, and if $\text{Mult}(A_{m+1}, i) < \text{Mult}(A_m, i)$ then by the interlacing property above the multiplicity of i must reach $\text{Mult}(A_m, i)$ again on the way to $\text{Mult}(A, i)$ (here it is critical that $a_i(m) \neq 0$ only if $\text{Mult}(A_m, i) \leq \text{Mult}(A, i)$).
- The value $a_i(n)$ is (by definition) the multiplicity of i as an eigenvalue of A . In particular, by Lemma 1 we have $\sum_i a_i(n) \geq \frac{3n}{4}$.
- As noted above, by interlacing and discrete continuity, each multiplicity between 0 and $a_i(n)$ is achieved at some point during the augmentation process. So for each j between 1 and $a_i(n)$, there is a unique m with $a_i(m) = j$, and these m are increasing in j for each i .

Definition 8.2.1. We will refer to the collection of sequences a_i corresponding to a matrix A as the *type* of the matrix.

Note that the number of possible types for an integral matrix is not too large. There are at most $(n+1)^{4\sqrt{n+1}}$ possible choices for the $a_i(n)$ (since each $a_i(n)$ is an integer between 0 and n). Once the $a_i(n)$ are chosen, there are at most $n^{a_i(n)}$ possible values for each a_i

(choosing where each nonzero value in the vector is). Multiplying over all i , the number of possible types given $a_i(n)$ is at most

$$n^{\sum_i a_i(n)} \leq n^n.$$

So the total number of distinct possible types for A is at most $(n+1)^{4\sqrt{n}+1}n^n = 2^{o(n^{3/2})}$.

This means it is enough to show:

Claim 8.2.2. For any fixed type, the probability G is both integral and has that type is at most $2^{-cn^{3/2}}$.

So let us now consider a fixed type. For $0 \leq m \leq n-1$, let E_{m+1} be the event that for every eigenvalue i with $a_i(m) \neq 0$, the multiplicity of i as an eigenvalue increases from $a_i(m)$ to $a_i(m)+1$ as we augment A_m to A_{m+1} . These events correspond to the each eigenvalue's multiplicity following the correct track throughout the augmentation process. For a matrix to have the desired type, each of the E_{m+1} must hold, so we have

$$\begin{aligned} \mathbf{P}(A \text{ has the given type}) &\leq \prod_{m=0}^{n-1} \mathbf{P}(E_{m+1} | E_1, \dots, E_m) \\ &\leq \prod_{m=0}^{n-1} \max_{A_m} \mathbf{P}(E_{m+1} | A_m), \end{aligned}$$

where the last inequality follows since A_m completely determines the events E_1 through E_m .

By lemma 8.1.3, we know that for E_{m+1} to hold, x_{m+1} must be orthogonal to the eigenspace of A_m corresponding to each eigenvalue whose multiplicity increases from A_m to A_{m+1} . In particular, it must be simultaneously orthogonal to the eigenspace for every i where $a_i(m) \neq 0$, since by definition that corresponds to an increase in the multiplicity

of i (there may be other eigenvalues whose multiplicities increase but for which $a_i(m) = 0$ because m is not maximal for its multiplicity. We ignore them).

Since we are now conditioning on A_m , this corresponds to a fixed subspace in which x_{m+1} must lie for E_{m+1} to hold. That subspace has co-dimension equal to $\sum_i a_i(m)$. Using Lemma 8.1.4 the probability x_{m+1} lies in this subspace, we have

$$\mathbf{P}(E_{m+1}|A_m) \leq 2^{-\sum_i a_i(m)}.$$

Multiplying over all m and taking logarithms, we have

$$-\log_2(\mathbf{P}(A \text{ has the given type})) \geq \sum_{m=0}^{n-1} \sum_i a_i(m)$$

One way of thinking about this sum is that every time an eigenvalue's multiplicity increases, it contributes to the total co-dimension (and thus to the exponent) an amount equal to its current multiplicity. Recall that for each i and for each $0 \leq j \leq a_i(n)$, there is a unique m satisfying $a_i(m) = j$. So we can rewrite this bound as

$$\begin{aligned} -\log_2(\mathbf{P}(A \text{ has the given type})) &\geq \sum_{i=-2\sqrt{n}}^{2\sqrt{n}} \sum_{j=0}^{a_i(n)-1} j \\ &= \frac{1}{2} \sum_{i=-2\sqrt{n}}^{2\sqrt{n}} a_i(n)^2 - \frac{1}{2} \sum_{i=-2\sqrt{n}}^{2\sqrt{n}} a_i(n) \\ &\geq \frac{1}{2(4\sqrt{n}+1)} \left(\sum_{i=-2\sqrt{n}}^{2\sqrt{n}} a_i(n) \right)^2 - \frac{1}{2} \sum_{i=-2\sqrt{n}}^{2\sqrt{n}} a_i(n). \end{aligned}$$

By Lemma 8.1.1, we have

$$-\frac{3n}{4} \leq \sum_{i=-2\sqrt{n}}^{2\sqrt{n}} a_i(n) \leq n.$$

So it follows that

$$-\log_2(\mathbf{P}(A \text{ has the given type})) \geq \frac{1}{2(4\sqrt{n}+1)} \left(\frac{3n}{4}\right)^2 - \frac{n}{2} = \left(\frac{9}{128} + o(1)\right)n^{3/2}$$

as desired.

Chapter 9

Conclusions

9.1 Information Gathering in Ad-Hoc Radio Networks having Maximum Degree Δ

Our result for the question of aggregating information on trees with unknown topology having maximum depth D and maximum degree Δ has been essentially resolved. We cannot claim to have completely resolved the question since the behavior of $s_k(n)$ is not well understood when k is fixed and n is allowed to grow. We do conjecture that we have resolved the problem, which in terms of $s_k(n)$ can be expressed in the following way:

Conjecture 9.1.1. Let $s_k(n)$ be the size of a k -selector on n vertices. Let $c < 1$, then

$$\frac{s_k(n)}{s_k(n^c)} = \mathcal{O}(1)$$

Should conjecture 9.1.1 be true, then this thesis has successfully resolved the question.

9.2 Matrices Having Spectra of Fixed Type

The proof of Theorem 8.0.2 did not rely on the fact that the fixed spectrum considered were integers. An identical argument would hold that for any fixed spectrum S we would have the following bound.

Theorem 9.2.1. Let S be any subset of the algebraic integers, and suppose there are constants $\alpha < 2$ and c such that for sufficiently large N

$$|S \cap [-N, N]| \leq cN^\alpha$$

then there is a constant c' such that for sufficiently large n , the proportion of graphs on n vertices having spectrum lying entirely in S is at most $2^{-c'n^{2-\alpha/2}}$.

Similarly, although $(0, 1)$ matrices were natural to look at due to the graphical motivations, the actual distribution of the entries was not critical here. A similar theorem would hold if the entries were drawn from any other bounded non-degenerate distribution (with the constant now depending on the distribution in question).

It seems unlikely that $3/2$ is the correct exponent in Theorem 8.0.2, and indeed we suspect that the probability a graph is integral is $2^{-(\frac{1}{2}+o(1))n^2}$ (equivalently, that the number of integral graphs is $2^{o(n^2)}$). However, it seems that improving the exponent beyond $3/2$ will require some significant new idea. The main stumbling block is in a sense estimating the number of graphs for which a fixed eigenvalue appears with large multiplicity. Even in the case $\lambda = 0$, this seems like an interesting problem.

Question 9.2.2. Let Q_n be a random $n \times n$ symmetric matrix where each above diagonal entry is equally likely to be 0 and 1. For a given s (possibly growing with n), what is the probability Q_n has rank at most $n - s$?

In the case $s = 1$, this corresponds to estimating the singularity probability of Q_n , which is a well-studied problem [9, 17, 20]. The current best upper bound is due to Vershynin, who showed in [20] that for large n the probability is at most $\exp(-n^c)$ for some $c > 0$. The best known lower bound is $(1 + o(1))\binom{n}{2}2^{-n}$, coming from the probability that some pair of rows of Q_n are equal (and it is a longstanding conjecture that this bound is optimal).

For larger s , the authors of [1] showed an upper bound of $2^{-\frac{s^2-s}{2}}$ in their proof of Theorem 8.0.1, and a similar bound showed up in our argument where we estimated the probability of a matrix having a given type (with $a_i(n)$ for a single i playing the role of s). A natural lower bound here would be the probability that Q_n contains at least s zero rows (equivalently, that the corresponding graph has at least s isolated vertices), which for s much smaller than n is $2^{-(1+o(1))ns}$. This bound may well be essentially optimal.

Bibliography

- [1] O. Ahmadi, N. Alon, I. Blake, and I. Shparlinski, Graphs with integral spectrum. *Linear Alg. Appl.* **430** (2009), 547-552
- [2] R. Bar-Yehuda, O. Goldreich, and A. Itai, On the time complexity of broadcast in radio networks: An exponential gap between determinism and randomization, *Journal of Computer and System Sciences* **45** (1992) 104-126
- [3] M. Bašić, M. Petković, and D. Stevanović, Perfect state transfer in integral circulant graphs, *Appl. Math. Lett.* **22** (2009), 1117-1121
- [4] Chlamtac, I., Kutten, S.: On broadcasting in radio networks - problem analysis and protocol design. *IEEE Transactions on Communications* **33** (1985), 1240-1246
- [5] M. Christandl, N. Datta, A. Ekert, and A.J. Landahl, Perfect state transfer in quantum spin networks, *Phys. Rev. Lett.* **92** (2004), 189702
- [6] M. Chrobak, K. Costello, L. Gasieniec, and D. R. Kowalski, Information gathering in ad-hoc radio networks with tree topology, In Proc. 8th International Conference on Combinatorial Optimization and Applications (COCOA14), (2014) 129-145
- [7] M. Chrobak, L. Gasieniec, and W. Rytter, Fast broadcasting and gossiping in radio networks, *J. Algorithms* (2002), 43(2):177-189
- [8] A. Clementi, A. Monti, and R. Silvestri, Distributed broadcast in radio networks of unknown topology, *Theor. Comput. Sci.*, **302**(1-3) (2003) 337-364
- [9] K. Costello, T. Tao, V. Vu, Random symmetric matrices are almost surely non-singular, *Duke Math J.* **135** (2006), no. 2, 395-413
- [10] K. Costello, P. Williams, On the Number of Integral Graphs, *Linear Algebra and its Applications* **493** (2016) 447-454
- [11] P. Erdős, P. Frankl, and Z. Füredi, Families of finite sets in which no set is covered by the union of r others, *Israel Journal of Mathematics*, **51**(12), (1985)

- [12] R. Holte, A. Mok, L. Rosier, I. Tulchinsky, D. Varvel, Pinwheel: a real-time scheduling problem, Proceedings of the Twenty-Second Annual Hawaii International Conference on System Sciences. Volume II: Software Track (1989)
- [13] R. Horn and C. Johnson, Matrix Analysis, 2nd edition, Cambridge University Press, (1999)
- [14] F. Harary and A.J. Schwenk, Which Graphs have Integral Spectra? *Graphs and Combinatorics*. Lecture Notes in Mathematics Vol. 406 (1974) 45-51
- [15] J. Komlós, On the determinant of $(0, 1)$ matrices, *Studia. Sci. Math. Hungar.* **2** (1967), 7-22
- [16] G. De Marco, Distributed broadcast in unknown radio networks, In Proc. 19th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA08)
- [17] H Nguyen, Inverse Littlewood-Offord problems and the singularity of random symmetric matrices, *Duke Math J.* **161** (2012), 545-586
- [18] A. Odlyzko, On subspaces spanned by random selections of ± 1 vectors, *J. Comb. Theory A*, **47** (1988) 124-133
- [19] T. Tao and V. Vu, Random matrices have simple spectrum, *Preprint*, available at arXiv:1412.1438
- [20] R. Vershynin, Invertibility of symmetric random matrices, *Random Structures and Algorithms* **44** (2014), 135-182
- [21] E. Wigner, On the distribution of the roots of certain symmetric matrices, *Ann. Math.* (2) **67** (1958), 325-327