1995

Cayley graphs as models for parallel processing supercomputer architectures

Karl Yorston
San Jose State University

Recommended Citation
Yorston, Karl, "Cayley graphs as models for parallel processing supercomputer architectures" (1995). *Master's Theses*. 1043.
DOI: https://doi.org/10.31979/etd.ffqe-e5kr
https://scholarworks.sjsu.edu/etd_theses/1043

CAYLEY GRAPHS AS MODELS FOR PARALLEL PROCESSING SUPERCOMPUTER ARCHITECTURES

A Thesis

Presented to the Faculty of

The Department of Mathematics and Computer Science

San Jose State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Karl Yorston

May 1995

APPROVED FOR THE DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE

Roger Alperin

Dr. Roger Alperin

Dr. Brian Peterson

Bradley W. Jackson

Dr. Bradley W. Jackson

APPROVED FOR THE UNIVERSITY

Serena W. Stanford

ABSTRACT

CAYLEY GRAPHS AS MODELS FOR PARALLEL PROCESSING SUPERCOMPUTER ARCHITECTURES

by Karl Yorston

This thesis traces efforts in recent literature to construct graphs -- principally Cayley graphs -- with desirable characteristics as models for parallel processing supercomputer architecture. Bounds on the diameters and expansion constants of such graphs are found, particularly through the use of eigenvalues of associated matrices.

While construction of exotic Cayley graphs with relatively low diameters proved possible, it was generally achieved at the expense of efficient routing algorithms and flexibility in design, leaving doubt as to whether these exotic graphs improve upon the benchmark hypercube. This thesis develops an alternate approach of Cayley graphs based on the common group Z_n. Such groups provide for a high degree of design flexibility and lead to Cayley graphs that are of practical improvement on the hypercube architecture.

ACKNOWLEDGMENT

My thanks go to my thesis advisor, Dr. Roger Alperin, for his patient explanations and support during this undertaking. I am also grateful to Dr. Bradley W. Jackson and Dr. Brian Peterson for agreeing to wade through this lengthy tome: I hope that they felt it worth the effort.

I wish also to express my appreciation to the SJSU Math and Computer Science Department faculty and the "superstaff" of the department office, the former for their high standard of math instruction both in and out of class, and the latter for their patient aid in navigating the murky waters of university policy and paperwork.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

Z_n	cyclic group of order n
S_n	the permutation group on n letters
Γ	group (generic)
S	generating set of a group
$G = G(V,E)$	graph
$V(G)$	vertex set of graph G
$E(G)$	edge set of graph G
$e = \{u,v\}$	edge of G between vertex u and vertex v
$\deg(v)$	degree of vertex v
$d = \deg(G)$	(maximum) degree of graph G
$u\text{-}v$	path from u to v in a graph
$d(A,B)$, $\rho(A,B)$	distance from set A to set B in a graph
$d(u,v)$	distance from vertex u to vertex v
$\mathrm{Diam}(G)$	diameter of graph G
$\mathrm{AlgDiam}(G)$	algorithmic diameter of graph G
$A(G)$	standard adjacency matrix of graph G
$K = K(G)$	diagonal matrix of G, $K_{ii}=\deg(v_i)$, $v_i \in V$
$Q = Q(G)$	matrix of G, where $Q = K-A(G)$
$\lambda_1 = \lambda(G)$	smallest nonzero eigenvalue of Q
$\mathrm{spec}(M)$	spectrum (set of eigenvalues) of M
μ	spectral radius of A(G)
$m(\xi)$	multiplicity of eigenvalue ξ
E_A	set of edges with both ends in set A
$E(A,B)$	bridge of sets A and B
$N(A)$	neighborhood of set A
∂A	boundary of set A
$\kappa_\Gamma(S,\pi), \kappa_\Gamma(S), \hat{\kappa}_\Gamma(S)$	various Kazhdan constants (Def. 3.4.1)
$h(G)$	Cheeger constant of G
$c_{max} = c_{max}(G)$	expanding constant of G
$c = c(G)$	magnifying constant of G
(f,g)	vector product of f and g
$C(\ ,\ )^{*}$	combination
$\lfloor a \rfloor$	floor function of a

CHAPTER 1

GRAPH MODELLING OF SUPERCOMPUTER ARCHITECTURE

1.1 Desirable Architectures and Graphs

In recent years, we have all come to expect regular increases in the speed and power of computers. For this trend to continue, many computer experts believe that we must continue to develop the design of massively parallel processing supercomputers. Such computers depend on the concept of many relatively simple processors connected by a network architecture and instruction set designed to take advantage of their simultaneous calculation power.

For such a large number of processors, designing a desirable interconnection network between them is a formidable challenge. When a processor is finished with a particular task, it must pass the resulting information on to another processor which will further manipulate that information. It would, of course, be ideal if the "next" processor in this sequence of operations were directly wired to the "previous" processor. However, computers are designed to perform many different tasks, so it would severely restrict a computer's usefulness to be wired into a particular processing sequence since the "next" operation required will vary from task to task. Thus, to enhance the versatility of the computer, it must be wired in such a way that any processor can send information to any other processor.

Unfortunately, with many processors the cost in materials, labor, and service is prohibitively high when wiring every processor directly to every other processor. In addition, it may be physically nearly impossible to wire it up in such a fashion. However, if information must often be passed through many processors to reach its next processing destination, valuable clock ticks are wasted in this "hand-off" process, thus slowing the computer's speed at accomplishing a task.

Clearly, then, we have a trade-off between how closely connected the processors are to each other and how expensive and difficult the wiring job will be. If a processor has more processors hard-wired to it then the chances improve that a given processor will be reached in fewer information hand-offs; that is, the "distance" (the number of hand-offs required) between any two processors should decrease on average. However, it seems apparent that a clever design for the processor network might reduce the maximum (and thus, logically, the average) number of hand-offs required between any two processors without increasing the number of interconnections. Thus, smarter wiring should lead to less-expensive and faster computing.

Hence, for a fixed number of processors, it is desirable to keep both the number of wires connected to each processor (the **degree** of the processor) and the maximum and average distances between any two processors (the **diameter** and **average diameter** of the network, respectively) as small as possible. In addition, it is much easier to construct and program such a computer if it is "vertex-symmetric"; that is, if the network structure is the same as viewed from any processor. Among other things, this requires that the degree of processors in the network be regular, meaning that the degree of each network processor is the same.

Reducing the wiring diagram of such a computer architecture to its simplest form would yield a drawing of dots for processors, with the lines connecting those dots representing the wires between processors. Such a figure in mathematics is called a graph, which we define with the help of Chartrand and Lesniak [CL] as follows:

DEFINITION 1.1.1: A **graph G** is a finite nonempty set of objects called **vertices** together with a (possibly empty) set of unordered pairs of distinct vertices called **edges**. The **vertex set** of G is denoted by V(G), while the **edge set** of G is denoted by E(G). If e={u,v} is an **edge** of graph G, then u and v are **adjacent vertices**, while u and e are **incident**, as are v and e. A graph is called **simple** if it has no loops (no vertex is adjacent to itself) and no multiple edges between pairs of vertices. □

If, as is customary, a (simple) graph is drawn by a diagram of dots or circles for vertices and line segments for edges, we immediately see its usefulness in depicting the wiring "blueprint" of a processor network. Then we may use such representations to bring the tools of graph theory into the study of processor networks. Of course, the concepts of degree, distance, and diameter in a processor network have their counterparts in graph theory.

DEFINITION 1.1.2: The **degree of a vertex v**, denoted by deg(v), in a graph G is the number of edges of G incident with v. The **degree of a graph G** is the maximum value of the degree of any vertex in G. If the degree of each vertex in G is equal to the same constant k, then G is a **k-regular** graph. □

Clearly, in a graph representing a processor network, the degree of a vertex is the same as the number of wires connected to its represented processor which lead to other processors. Similarly, the distance between any two processors in a network has an analogous definition.

DEFINITION 1.1.3: A **u-v path** in a graph G is a sequence of vertices alternating with incident edges which traces a connected route in G that begins with vertex u, ends with vertex v, and repeats no vertex (and hence no edge). The **length** of a path is the number of edges listed in the sequence of that path. The **distance d(u,v)** between vertices u and v in graph G is the minimum length of all possible u-v paths in G. If no such connected route exists in G, the distance is given to be infinite. □

Thus, the distance between any two processors is the shortest path through the graph of that network between their corresponding pair of vertices. The diameter of a network and its modelled graph follow in a natural way.

DEFINITION 1.1.4: The **diameter of a graph G**, denoted by Diam(G), is the maximum distance found in the set of distances between all possible pairs of vertices in G. The **average diameter of G** is the sum of the distances between all possible pairs of vertices in G divided by the number of such pairs. □

With these definitions taken care of, it is possible to formalize the definition of vertex-symmetry for graphs.

DEFINITION 1.1.5: A graph G is **vertex-symmetric** if the automorphism group on the graph G acts transitively on the vertex set of G. □

Since any element in the automorphism group on the graph G must preserve the structure (i.e., adjacencies) of G, we see that the ability of an automorphism group to act transitively on the entire vertex set of G would fulfill our desire to construct a network that looks the same when viewed from each vertex.

Finally, it is important to recognize that computers send information between processors using a routing algorithm. Since an exhaustive search of the shortest path between processors is extremely cumbersome, the distances found in the graph representation of a network may in fact be shorter than the actual algorithmic routes taken by information sent from one processor to another. We thus expand on Definition 1.1.4 to state that the **algorithmic diameter** of a network with respect to its routing algorithm is the maximum length of any route taken by the algorithm between any pair of processors in the network. The **average algorithmic diameter** is the corresponding average of such routes, and is again algorithm dependent.

Hence, our goal is to find vertex-symmetric graphs with small degree, diameter, and average diameter which model computer architectures with efficient routing algorithms leading to algorithmic full and average diameters as close to the true graph diameters as possible.

1.2 Cayley Graphs

In the last decade or so, much research has been devoted to the structure resulting from modeling a processor network with a graph whose underlying vertex set is composed of a group. This construction allows the use of various techniques in graph theory, group theory, and linear algebra to look at the properties of such graphs. In particular, the constraint of structural symmetry on the search for a clever processor network has usually steered the choice of candidates to the so-called Cayley graphs.

The following definition and theorem show that Cayley graphs possess the requisite vertex-symmetric property.

DEFINITION 1.2.1: Let Γ be a group and let $S=S^{-1}$ be a generating set for Γ which is closed under inverses and does not contain the identity. The **Cayley graph of Γ with respect to S** is the graph $G=G(V,E)$ with vertex set $V=\Gamma$ and edge set

$$E = \{\{v_1,v_2\} \mid s_i{\cdot}v_1 = v_2 \text{ for some } s_i\epsilon S\} \qquad \square$$

Note that such a graph is clearly $|S|$-regular with no loops, since each vertex will have one edge leaving it for each element in S and the restriction that S does not contain the identity prevents any of those edges from returning to that same vertex. In addition, it follows that G is a connected graph, since S is a generating set for the group Γ, thus ensuring that a path exists from any vertex to any other vertex.

Figure 1.2.2 shows the Cayley graph on S_3 with respect to the generating set $S=\{a,a^{-1},b\}$ where $a=(123)$, $a^{-1}=(132)$, and $b=(12)$. Observe that, by convention, an edge is given a

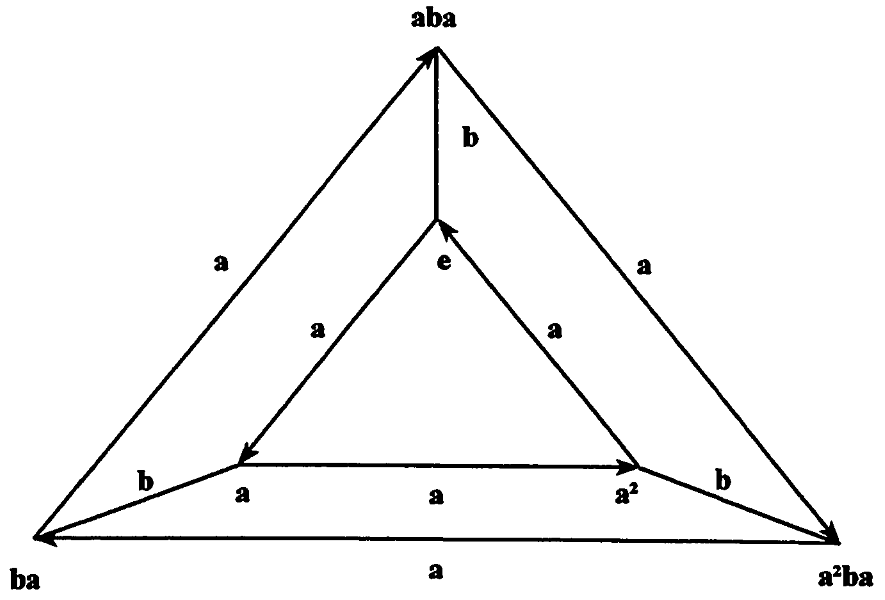

Figure 1.2.2: Cayley graph of S_3 with respect to the generating set $S=\{a,a^{-1},b\}$, where $a=(123)$ and $b=(12)$.

direction to indicate the movement resulting from left multiplication if the generator is not its own inverse, while it is not given an arrow if the generator is its own inverse. Inspection of the Cayley graph in Figure 1.2.2

tends to confirm the belief that Cayley graphs are vertex-symmetric. Theorem 1.2.3 shows conclusively that this is indeed the case.

THEOREM 1.2.3 ([SS], Proposition 1.1): Let Γ be a group. Then the Cayley graph G on Γ with respect to a generating set S is a vertex-symmetric graph.

***Proof*:** From Definition 1.1.5, we must show that the automorphism group of G acts transitively on the vertex set V(G). That is, we must be able to reach every vertex from any other through such automorphisms.

Let S_Γ be the permutation group on the elements of Γ.

Then, for $h \epsilon \Gamma$, we define the permutation $\sigma_h \epsilon S_\Gamma$ by

$$\sigma_h(x) = x * h \text{ for every } x \epsilon \Gamma.$$

σ_h will be an automorphism of G if its relabeling preserves adjacency. That is, we must demonstrate that if x and y are adjacent in G, then $\sigma_h(x)$ and $\sigma_h(y)$ were also adjacent in G. We reason as follows:

Two vertices x,y in G are adjacent

iff the edge connecting them is in the generating set S

iff $\{x,y\} \in E(G)$

iff there exists $s_i \in S$ such that $s_i * x = y$

iff $y * x^{-1} \in S$.

But

$$\begin{aligned} y*x^{-1} = y*(h*h^{-1})*x^{-1} &= y*h*(h^{-1}*x^{-1}) \\ &= y*h*[x*h]^{-1} \\ &= \sigma_h(y)*\sigma_h(x)^{-1}. \end{aligned}$$

Hence, we have that x and y are adjacent in G if and only if $\sigma_h(x)$ and $\sigma_h(y)$ are adjacent in G, demonstrating that σ_h preserves structure and is therefore in the automorphism group of G.

Then, for any pair of vertices x and y in G, the permutation $\sigma_{x^{-1}*y}$ is in the group of automorphisms of G since

$x^{-1} * y \in \Gamma$, and we have

$$\sigma_{x^{-1} * y}(x) = x * (x^{-1} * y) = y.$$

Thus, we find that the automorphism group on G is transitive on the vertex set V(G), and hence the Cayley graph is vertex-symmetric. □

Since any finite group may be the basis for a Cayley graph, we have an infinite source of vertex-symmetric graphs available. In addition, any generating set (without the identity) will do, so each group alone may produce a wide range of alternative graphs.

According to [SS], most of the current large-scale parallel-processing computers are based on a vertex-symmetric architecture. They mention the 12-dimensional binary hypercube of the Connection Machine, the 256×256 torus-connected 2-dimensional mesh for the MPP at NASA/Goddard, the butterfly network, and the cube-connected cycle network as widely accepted models for network architectures, all of which are vertex-symmetric graph structures.

1.3 The Hypercube Design

The standard against which to compare the performance of proposed architectures is often that of the binary hypercube (or n-cube). The underlying Cayley graph for this computer is based on the group $\Gamma = Z_2 \times Z_2 \times \ldots \times Z_2$ of order 2^k and generating set $S=\{(1,0,\ldots,0), \ldots, (0,0,\ldots,1)\}$. Since each generator is its own inverse, S is closed under inverses and is of order $|S|=k$. The hypercube may be visualized as a k-dimensional cube -- each vertex being a "corner" -- of degree k, diameter k, and average diameter k/2.

However, since we are dealing with a computer application, we again note the importance of finding the values of the algorithmic diameter and the average algorithmic diameter, since these will provide a more realistic appraisal of how fast and efficient an architecture's routing algorithms will be. Each vertex in the hypercube is named with a binary string, so determining a path between two vertices is as simple as subtracting the starting vertex's value from that of the target vertex. The resulting "road map" is a binary number where every nonzero entry indicates a generator's edge that is necessary to travel in order to reach the target vertex.

Because of this simplicity in the hypercube's routing, the algorithmic diameter and average diameter match the theoretical k and k/2, respectively. In addition, any edge with a corresponding 1 in the binary road map may be taken at any step. This offers convenient alternative routing to avoid bottlenecks in information transfer. This perfect and bottleneck-free routing algorithm makes the hypercube an attractive design.

However, the hypercube does have its disadvantages. First, the degree is rather high for the number of processors used. This results in high wiring costs. In addition, [SS] state that vertex degrees above 6 may cause data path bandwidth problems for switches in the network, though their argument is perhaps questionable and in any event may not reflect current technology.

Second, the hypercube may only be constructed on 2^k vertices. Hence the choices for network size and wiring are quite limited, and adapting an existing machine to a hypercube of a different size would be quite difficult and expensive.

In this thesis, we survey both theoretical and practical efforts to provide graphs with improved properties as compared to the hypercube. Chapters 2 through 4 follow several researchers' theoretical work to develop bounds on diameters and expansion-related constants of various graphs. Chapter 5 examines a routing algorithm proposed in [SS], and considers its impact on diameter and average diameter. Finally, Chapter 6 contains a study of Cayley graphs on groups of class Z_n as candidates for supercomputer architecture, demonstrating that such networks are in many ways superior to the benchmark hypercube.

CHAPTER 2

RELATIONSHIPS ON DIAMETER TO EIGENVALUES

2.1 Introduction

This chapter evaluates two efforts to find an upper bound on the diameter of a connected graph G on n vertices as a function of certain eigenvalues. Chapter 2.2 will examine the work of Alon and Milman ([AM], Section 2) who focus on λ_1, the smallest nonzero eigenvalue of the matrix $Q=K-A(G)$, where K is the diagonal matrix whose entry $K_{ii}=\deg(v_i)$ and A(G) is the standard adjacency matrix of G. Chapter 2.3 will discuss the results of Chung [Ch] who focuses on the spectral radius μ of the standard adjacency matrix A(G), where G in this case is a connected k-regular graph. Chapter 2.4 will show the relationship of these two eigenvalues, while Chapter 2.5 demonstrates that both bounds are quite "soft" in terms of providing useful limits on the diameters of Cayley graphs of the cyclic groups Z_n.

2.2 Alon and Milman's work on λ_1

We first define and discuss the matrix Q used by Alon and Milman. Further discussion of the quadratic form on Q results in establishing some facts about the eigenvalues of Q, particularly about its smallest nonzero eigenvalue λ_1. We then follow Alon and Milman's development of several theorems built around characteristics of λ_1, the last theorem of which demonstrates an upper bound on the diameter of a graph.

The Matrix Q:

Let $G=(V,E)$ be a connected graph on $|V|=n$ vertices, and let D be an orientation of G. Let C be the $|E|\times|V|$ incidence matrix for the orientation D of G, so that it has $|E|$ rows indexed by the edges of D and $|V|$ columns indexed by the vertices of D. Note that

$$C_{e,v} = \begin{cases} 1 & \text{if v is the head of e; that is, if } v=e^{+}. \\ -1 & \text{if v is the tail of e; that is, if } v=e^{-}. \\ 0 & \text{otherwise.} \end{cases}$$

Then the i^{th} row of C consists of zeros except for a "+1" in the column corresponding to the vertex e_i^+ at the head of edge e_i, and a "-1" in the column corresponding to the vertex e_i^- at the tail of edge e_i. Hence, for f a real-valued function on the vertex set V (i.e., an n-tuple with real entries), we have that Cf is an $|E|\times 1$ "vector" (or $|E|$-tuple) whose i^{th} entry is

$$+f(e_i^+) - f(e_i^-).$$

Thus,

$$\begin{aligned} (Cf,Cf) &= [f(e_1^+)-f(e_1^-),\ldots,f(e_k^+)-f(e_k^-)] \\ &\qquad \bullet [f(e_1^+)-f(e_1^-),\ldots,f(e_k^+)-f(e_k^-)] \\ &= [f(e_1^+)-f(e_1^-)]^2 + \ldots + [f(e_k^+)-f(e_k^-)]^2 \\ &= \sum_{e\in E} [f(e^+)-f(e^-)]^2. \end{aligned}$$

Define $Q = C^T C$. Then Q is clearly an nxn matrix, since C^T is $n \times |E|$ and C is $|E| \times n$. In addition, $Q=K-A(G)$, where K is the diagonal nxn matrix such that $K_{ii}=\deg(v_i)$ and $A(G)$ is the standard nxn adjacency matrix of G. That is, we have

$$A(G)_{ij} = A(G)_{ji} = \begin{cases} +1 & \text{if} \quad (v_i, v_j) \in E \\ 0 & \text{otherwise.} \end{cases}$$

Note that the conventional "standard adjacency matrix" has zeros in the diagonal since a vertex is not considered adjacent to itself. That is, there are no "loops" at each vertex and the identity is not in the edge set of the graph. Hence, $A(G)_{ii}=0$, and the sum of the entries ("row sum") of row i is equal to the degree of vertex v_i.

Consider the entry Q_{ii}:

> It is formed from the dot product of row i of C^T and column i of C, which, of course, is equivalent to the dot product of column i of C with itself. Hence

$$Q_{ii} = \sum C_{ji}^2, \text{ where j runs from 1 through } m=|E|.$$

But from above, $C_{ji}=\pm 1$ whenever vertex v_i is adjacent to vertex v_j in G. Since v_i is adjacent to $\deg(v_i)$ vertices in G, there will be precisely $\deg(v_i)$ nonzero terms in the summation, each term of which is $(\pm 1)^2 = +1$. Hence, $Q_{ii}=\deg(v_i)$ for each $v_i \in V$.

Consider the entry Q_{ij} $(i \neq j)$:

It is formed from the dot product of row i of C^T and column j of C, which, of course, is equivalent to the dot product of column i of C with column j of C. Hence

$$Q_{ij} = \sum (C_{ki}C_{kj}), \text{ where k runs from 1 through } m=|E|.$$

Suppose for a given value of k that C_{ki} and C_{kj} are both nonzero entries. Then this implies that in row k of C there is a "+1" in either column i or column j, and a "-1" in the other. That is, edge e_k in D goes from v_i to v_j or from v_j to v_i. In either case, what this says is that v_i and v_j are adjacent to each other in G. Clearly, this can occur only once for each pair of i and j. Hence, Q_{ij} can have a nonzero term in its sum at most once, that occurring only when v_i and v_j are adjacent in G. Thus, for all $i \neq j$, we have

$$Q_{ji} = Q_{ij} = \begin{cases} -1 & \text{if } v_i \text{ and } v_j \text{ are adjacent in G} \\ 0 & \text{otherwise.} \end{cases}$$

Clearly, then, Q may be separated into the difference of the nxn diagonal matrix K and the nxn adjacency matrix A(G). In addition, this result is independent of the orientation D chosen since both K and A(G) are only functions of G. Finally, Q is symmetric, since it is the sum of two symmetric matrices. (Any diagonal matrix K is by definition symmetric, and A(G) is symmetric since if $A(G)_{ij}=1$, then v_i is adjacent to v_j in G and therefore v_j is adjacent to v_i in G, and so $A(G)_{ji}=1$ as well.)

The Quadratic Form on Q (For background on inner product spaces, bilinear forms, and quadratic forms, see [FIS]):

Consider the set of functions that assign real values to each vertex. Such functions g and f may thus be represented as n-tuple with real entries. That is, $g,f \in \mathbb{R}^n$. Since Q is a symmetric, real-valued nxn matrix, we may consider the symmetric bilinear form $H: \mathbb{R}^n \times \mathbb{R}^n \mapsto \mathbb{R}$ defined by $H(g,f) = g^T Q f$. The associated quadratic form is defined as

$$\begin{aligned} K(f) = H(f,f) &= f^T Q f \\ &= f^T Q^T f, \text{ since Q is symmetric so } Q=Q^T, \\ &= (Qf)^T \bullet f \\ &= (Qf, f), \text{ since Qf is simply a vector in } \mathbb{R}^n. \end{aligned}$$

A quadratic form is called **positive semi-definite** if

$$K(f) = H(f,f) \geq 0 \text{ for every } f \in \mathbb{R}^n.$$

It is called **positive definite** if

$$K(f) = \begin{cases} H(\mathbf{0},\mathbf{0}) = 0 \text{ only when } f=\mathbf{0} \in \mathbb{R}^n \\ H(f,f) > 0 \text{ whenever } \mathbf{0} \neq f \in \mathbb{R}^n. \end{cases}$$

In our case, we have

$$
\begin{aligned}
(Qf,f) &= f^TQf = f^TC^TCf \\
&= (Cf)^T\bullet(Cf) \\
&= (Cf,Cf) \\
&= \textstyle\sum_{e\in E} [f(e^+)-f(e^-)]^2 \\
&\geq 0,
\end{aligned}
$$

since it is the sum of the squares of real numbers.

Hence, $(Qf,f) \geq 0$ for every $f\in\mathbb{R}^n$. Observe that if f is a constant function (i.e., an n-tuple with all identical entries) we have $Cf=\mathbf{0}$, since $f(e^+)-f(e^-) = 0$ for every $e\in E$, and so $(Qf,f) = (Cf,Cf) = (\mathbf{0},\mathbf{0}) = 0$. Therefore, we have that the quadratic form (Qf,f) is positive semi-definite.

Note also that if f is **not** a constant function, then for at least one pair of vertices v_i and v_j we have that $f(v_i)\neq f(v_j)$. Because D is connected, there exists a chain of edges between v_i and v_j in D, and thus we have that

$$Cf = [f(e_1^+)-f(e_1^-),\ldots,f(e_{|E|}^+)-f(e_{|E|}^-)] \neq \mathbf{0}$$

since eventually two end-vertices of an edge e_k in the chain must have different values assigned by f in order for $f(v_i) \neq f(v_j)$ to be true. Then Cf will have at least one nonzero entry corresponding to edge e_k, and, therefore,

$$(Qf,f) = (Cf,Cf) = \sum [f(e_i^+)-f(e_i^-)]^2 \geq [f(e_k^+)-f(e_k^-)]^2 > 0.$$

Thus, if f is a nonconstant function that is an eigenvector of Q in $\mathbb{R}^n$, the corresponding eigenvalue λ_f must be positive, since

$$\begin{aligned} 0 < (Qf,f) = (\lambda_f \cdot f,f) &= \sum(\lambda_f[f(e_i^+)-f(e_i^-)])[f(e_i^+)-f(e_i^-)] \\ &= \sum\lambda_f \ [f(e_1^+)-f(e_1^-)]^2 \\ &= \lambda_f\sum[f(e_1^+)-f(e_1^-)]^2. \end{aligned}$$

A well-known theorem (see [FIS], Theorem 6.29, p 362) states that, for a finite dimensional vector space V over a field F not of characteristic two, every symmetric bilinear form on V (and hence each corresponding quadratic form on V) is diagonalizable. Since any symmetric matrix $A \in M_{nxn}(F)$ is congruent to its diagonal matrix and thus shares the same eigenvalues, we know that Q has n eigenvalues, with each appearing according to its multiplicity. Clearly, 0 appears only once as an eigenvalue, since the subspace of all constant functions has dimension 1. By the above analysis, all the remaining n-1 eigenvalues are positive. Hence, we have as the eigenvalues of Q

$$0 = \lambda_0 < \lambda_1 = \lambda(G) \leq \lambda_2 \leq \ldots \leq \lambda_{n-1}.$$

THEOREM 2.2.1 ([AM], Lemma 2.1): Let $G=G(V,E)$ be a connected graph on $|V|=n$ vertices. Let A and B be two disjoint subsets of the vertex set V, and ρ be the distance (in G) between the set A and the set B. That is

$$\rho = min\{d(v_a,v_b) : v_a \in A,\ v_b \in B\},$$

where $d(v_a,v_b)$ is the length of the shortest path between vertex v_a and vertex v_b. Then,

$$\lambda_1 n \leq \rho^{-2}(1/a + 1/b)(|E|-|E_A|-|E_B|),$$

where $a=|A|/n$ (or $b=|B|/n$), the fraction of total vertices contained in A (B); and E_A (E_B) is the set of edges with both endpoints in A (B).

***Proof*:** Define a function (n-tuple) $g \in \mathbb{R}^n$ by

$$g(v) = 1/a - [(1/\rho)(1/a + 1/b) \cdot min(\rho(v,A),\rho)],$$

where $\rho(v,A) = min\{d(v,a_i): a_i \in A\}$, the distance from vertex v to set A. Hence, g is a nonconstant function, since it sends vertex v to one of the following values:

i) $1/a$, if $v \in A$, since then $\rho(v,A)=0$, implying that $[(1/\rho)(1/a + 1/b) \cdot min(\rho(v,A),\rho)] = 0$;

ii) $-1/b$, if $v \in B$, since then $\rho \leq \rho(v,A)$, and so we have $1/a - \rho^{-1}(1/a + 1/b)\rho = 1/a - 1/a - 1/b = -1/b$;

iii) a value between $1/a$ and $-1/b$ (inclusive) if v is in neither A nor B.

Observe that if two vertices u and v are adjacent, then

$$\begin{aligned}|g(u)-g(v)| &= |1/a - [\rho^{-1}(1/a + 1/b)\cdot \mathit{min}(\rho(u,A),\rho)] \\ &\qquad - (1/a - [\rho^{-1}(1/a + 1/b)\cdot \mathit{min}(\rho(v,A),\rho)])| \\ &= \rho^{-1}(1/a + 1/b)\cdot|\mathit{min}(\rho(v,A),\rho)-\mathit{min}(\rho(u,A),\rho)| \\ &\le \rho^{-1}(1/a + 1/b),\end{aligned}$$

since $|\mathit{min}(\rho(v,A),\rho)-\mathit{min}(\rho(u,A),\rho)| \le 1$ due to the adjacency of u and v implying that the difference in distance is at most one edge.

Define $\alpha=(1/n)\sum_{v\in V} g(v)$. Since there are n vertices to run through, α is simply the average of all the g(v)'s. Set $f=g-\alpha$. Then f is a nonzero function (an n-tuple) in $\mathbb{R}^n$, since we observed above that g is a nonconstant function. However, since the average value of g(v) is subtracted from each entry, the sum of the entries in f is thus zero. That is

$$\begin{aligned}\sum_{v\in V} f(v) = \sum_{v\in V} (g-\alpha)(v) &= \sum_{v\in V} [g(v)-\alpha] \\ &= [\sum_{v\in V} g(v)]-n\alpha \\ &= n\alpha-n\alpha = 0.\end{aligned}$$

Consider a constant (real-valued) function $k \in \mathbb{R}^n$, where $k(v)=c$ for every $v \in \mathbb{R}$. Then

$$\begin{aligned}(f,k) = f\bullet k &= [(g-\alpha)(v_1),(g-\alpha)(v_2),\ldots,(g-\alpha)(v_n)]\bullet(c,c,\ldots,c)\\ &= \sum_{v\in\mathcal{V}}[c\cdot(g-\alpha)(v)]\\ &= c\cdot\sum_{v\in\mathcal{V}}(g-\alpha)(v)\\ &= c\cdot 0 = 0.\end{aligned}$$

Hence, we have created a nonzero function $f \in \mathbb{R}^n$ that is orthogonal to the constant functions in $\mathbb{R}^n$. By a slight adaptation of the standard proof involving the Rayleigh quotient (see Appendix A) we have that $(Qf,f) \geq \lambda_1\|f\|^2$. Then we may argue as follows:

$$\begin{aligned}[(1/a^2) - 2\alpha/a + \alpha^2]a + [(1/b^2) - 2\alpha/b + \alpha^2]b\\ &= 1/a - 2\alpha + \alpha^2 a + 1/b + 2\alpha + \alpha^2 b\\ &= 1/a + 1/b + \alpha^2(a+b)\\ &\geq 1/a + 1/b\ , \text{ since } \alpha^2\geq 0,\ a>0, \text{ and } b>0.\end{aligned}$$

Hence, we clearly have that

$$\begin{aligned}\lambda_1 n(1/a + 1/b) &\le \lambda_1 n\{[(1/a^2)-2\alpha/a + \alpha^2]a + [(1/b^2)-2\alpha/b + \alpha^2]b\} \\ &= \lambda_1 n[(1/a - \alpha)^2 a + (1/b + \alpha)^2 b] \\ &= \lambda_1[(1/a - \alpha)^2 na + (1/b + \alpha)^2 nb].\end{aligned}$$

Obviously, $na=|A|$ and $nb=|B|$. In addition, $f(v)^2 = (1/a - \alpha)^2$ for every $v \in A$, while $f(v)^2 = [(-1)(1/b + \alpha)]^2 = (1/b + \alpha)^2$ for every $v \in B$. Hence,

$$\begin{aligned}\lambda_1 n(1/a + 1/b) &\le \lambda_1[\sum_{v\in A} f(v)^2 + \sum_{v\in B} f(v)^2] \\ &= \lambda_1 \sum_{v\in A\cup B} f(v)^2.\end{aligned}$$

Noting that $f(v)^2 \ge 0$ for every $v \in V$ and that $A\cup B \subset V$, we get

$$\lambda_1 \sum_{v\in A\cup B} f(v)^2 \le \lambda_1 \sum_{v\in V} f(v)^2,$$

since $A\cup B \subset V$, so at least as many vertices are counted in the right-hand sum, and thus

$$\lambda_1 \sum_{v\in A\cup B} f(v)^2 = \lambda_1\|f\|, \text{ since } \|f\| = f\bullet f = \sum_{v\in V} f(v)^2.$$

But

$$\begin{aligned}\lambda_1\|f\| &\le (Qf,f), \text{ by Rayleigh's Principle,}\\ &\le (Cf,Cf), \text{ by definition,}\\ &= \sum_{e\in E}[f(e^+)-f(e^-)]^2\\ &= \sum_{e\in E}[g(e^+)-g(e^-)]^2, \text{ since } f(v)=g(v)-\alpha,\\ &= \sum_{e\in E-(E_A\cup E_B)}[g(e^+)-g(e^-)]^2,\end{aligned}$$

since $g(e^+)-g(e^-)=1/a-1/a=0$ if $e\in E_A$ and $-1/b-(-1/b)=0$ if $e\in E_B$.

Recalling from above that

$$\rho^{-2}(1/a + 1/b)^2 \ge [g(e^+)-g(e^-)]^2 \text{ for every } e\in E,$$

then for the sum over $(|E|-|E_A|-|E_B|)$ terms we get

$$\sum_{e\in E-(E_A\cup E_B)}[g(e^+)-g(e^-)]^2 \le \rho^{-2}(1/a + 1/b)^2(|E|-|E_A|-|E_B|).$$

Hence

$$\lambda_1 n(1/a + 1/b) \le \rho^{-2}(1/a + 1/b)^2(|E|-|E_A|-|E_B|),$$

or

$$\lambda_1 n \leq \rho^{-2}(1/a + 1/b)(|E|-|E_A|-|E_B|),$$

as desired. □

THEOREM 2.2.2 ([AM], Theorem 2.5): Let all notations be as above, and let d be the maximum degree of a vertex of G. (d=k if G is k-regular). If $\rho>1$, then

$$b \leq (1-a)/[1+(\lambda_1/d)a\rho^2].$$

***Proof*:** Observe that $E-(E_A \cup E_B)$ is the set of all edges that don't have both endpoints in A or both endpoints in B. Hence, any edge in this set fits one of the following categories:

a) one end in A and the other in B;

b) one end in A and the other end is in neither A nor B;

c) one end in B and the other end is in neither A nor B;

d) both ends are non-A or non-B vertices.

But $\rho>1$, so case (a) is not possible since the distance between any vertex in A and any other vertex in B is greater than one, and therefore cannot be spanned by one edge. Thus, each edge contained in $E-(E_A \cup E_B)$ has at least one endpoint on a vertex in set $V'=V-(A\cup B)$, all the vertices in neither A nor B. Noting that $|V'| = |V|-|A|-|B| = n-na-nb$, and observing that the maximum degree in G is d, the most edges that could be incident with set V' is the maximum degree sum, which totals d(n-na-nb). Hence,

$$|E|-|E_A|-|E_B| \le n(1-a-b)d,$$

since E_A and E_B are disjoint. Combining with Theorem 2.2.1, we get

$$\begin{aligned}\lambda_1 n &\le \rho^{-2}(1/a + 1/b)(|E|-|E_A|-|E_B|)\\ &\le \rho^{-2}(1/a +1/b)[n(1-a-b)d]\\ &= \rho^{-2}[(b+a)/ab][n(1-a-b)d]\\ &\le \rho^{-2}[1/ab][n(1-a-b)d],\ \text{since } a+b \le 1.\end{aligned}$$

Cancelling n's and working further yields

$$
\begin{aligned}
& \rho^2 ab\lambda_1 \le d(1-a-b) \\
\Rightarrow \quad & b[1 + (\rho^2 a\lambda_1/d)] \le 1-a \\
\Rightarrow \quad & b \le (1-a)/[1 + (\lambda_1/d)a\rho^2]. \qquad \square
\end{aligned}
$$

THEOREM 2.2.3 ([AM], Theorem 2.6): Let all notation be as above. For $\rho \ge 1$ (not necessarily an integer) we have

$$b \le (1-a)\exp(-\mathit{ln}(1+2a)\lfloor(\lambda_1/2d)^2\rho\rfloor)$$

where exp(y), for real number y, represents e^y; and the symbol $\lfloor x \rfloor$, for real number x, is called the **floor function** and represents the greatest integer not exceeding x.

Proof: Since G is connected, the minimum degree of a vertex in G is greater than zero. Let u be a vertex of G with minimum degree. Suppose we were to set $A=\{u\}$, $B=V-\{u\}$. Then we have that: $\deg(u)=(|E|-|E_A|-|E_B|)$, since $|E_A|=0$ and all bridge edges are those leaving u; $\rho=1$, since the distance between A and B is clearly 1; and $a=1/n$ while $b=(n-1)/n$. Then we know from Theorem 2.2.1 that

$$\begin{aligned}\lambda_1 n &\le (n + 1/[(n-1)/n])\deg(u)\\ &\le [n + n/(n-1)]\deg(u)\\ &= n[1 + 1/(n-1)]\deg(u)\\ &= n[n/(n-1)]\deg(u),\end{aligned}$$

or

$$\lambda_1 \le [n/(n-1)]\deg(u).$$

Define $\eta=(2d/\lambda_1)^2$.

Since $d \ge \deg(u)$, we have $\lambda_1 \le [n/(n-1)]d$. For $n\ge 2$, we have $n/(n-1) \le 2$, and thus

$$\eta \ge (2d/[n/(n-1)d])^2 = (2/[n/(n-1)])^2 \ge (2/2)^2 = 1.$$

For a subset of vertices $F\subset V$ and positive real number r, define $F_r = \{v\in V\colon d(v,F)\le r\}$ as the set of all vertices in V within distance r of subset F. Set $k=\lfloor\rho/\eta\rfloor$. (Note that

$k \leq \rho$ since $\eta \leq 1$.) Then for integer j such that $0 \leq j < k$, we have that A_{ju} is the set of all vertices within distance ju of A, and $V-A_{(j+1)\eta}$ is the set of all vertices not within $(j+1)\eta$ of set A.

Hence, in $A_{(j+1)\eta}$ we have created a set with a "border" or "buffer" of width η around A_{ju}, thereby ensuring that, for all j, the distance between any vertex in A_{ju} and any vertex **not** in $A_{(j+1)\eta}$ is strictly greater than η. That is,

$$s_j = \mathit{min}\{d(v, V-A_{(j+1)\eta}) : v \in A_{ju}\} > \eta \geq 1.$$

In keeping with the notation of $a=|A|/n$, define $a_{ju}=|A_{ju}|/n$ and so $1-a_{ju} = |V-A_{(j+1)\eta}|/n$. Then, since the distance between such sets is at least one, Theorem 2.5 applies, and for every j such that $0 \leq j < k$ we have

$$\begin{aligned}
1 - a_{(j+1)\eta} &\leq (1-a_{ju})/[1 + (\lambda_1/d)a_{ju}s_j^2] \\
&\leq (1-a_{ju})/[1 + (\lambda_1/d)a_{ju}\eta^2], \text{ since } s_j^2 \geq \eta^2, \\
&\leq (1-a_{ju})/[1 + (\lambda_1/d)a\eta^2], \text{ since } a_{ju} \geq a, \\
&= (1-a_{ju})/[1 + (\lambda_1/d)a(2d/\lambda_1)], \text{ since } \eta^2=2d/\lambda_1,
\end{aligned}$$

$$= (1-a_{ju})/[1+2a].$$

Observing, then, that

$$1-a_{(0+1)\eta} \le (1-a_{0\eta})/[1+2a]$$

and multiplying both sides by the next inequality for j=1 we see

$$(1-a_{(1+1)\eta})(1-a_{(0+1)\eta}) \le \{(1-a_{1\eta})/[1+2a]\}(1-a_{0\eta})/[1+2a]$$

$$\Leftrightarrow \quad (1-a_{(1+1)\eta}) \le (1-a_{0\eta})/[1+2a]^2.$$

Continuing this process for each j up through k-1 yields

$$\begin{aligned} 1-a_{k\eta} &\le (1-a_{0\eta})/[1+2a]^k \\ &= (1-a)/[1+2a]^k, \text{ since } a_{0\eta}=a, \\ &= (1-a)\exp\{-ln(1+2a)k\} \\ &= (1-a)\exp\{-ln(1+2a)\lfloor\rho/\eta\rfloor\}, \text{ since } k=\lfloor\rho/\eta\rfloor, \\ &= (1-a)\exp(-ln(1+2a)\lfloor(\lambda_1/2d)^2\rho\rfloor). \end{aligned}$$

Since $B \subseteq V-A_{k\eta}$, we have that $b \le 1-a_{k\eta}$, thus yielding

$$b \le (1-a)\exp(-\mathit{ln}(1+2a)\lfloor(\lambda_1/2d)^2\rho\rfloor). \qquad \square$$

And, finally, we come to the theorem which provides an upper bound on the diameter of G.

THEOREM 2.2.4 ([AM], Theorem 2.7): Let $G=G(V,E)$ be a connected graph on $|V|=n > 1$ vertices, with maximal degree d. Then the diameter of G is at most

$$\mathrm{Diam}(G) \le 2\cdot\lfloor((2d/\lambda_1)^{1/2})\cdot\log_2 n\rfloor.$$

Proof: Suppose we have that $\rho = \eta\log_2 n = ((2d/\lambda_1)^{1/2})\cdot\log_2 n$. Without loss of generality, choose A to contain at least half of the vertices, so $|A|/n \ge 1/2$. Our first goal is to demonstrate in the notation of Theorem 2.2.3 above that for such a set A, $A_{\lfloor\rho\rfloor}=V$. That is, there are no vertices in V that are further than $\lfloor\rho\rfloor$ from set A. Hence, we shall show that the set $B=V-A_{\lfloor\rho\rfloor}$ must be the empty set.

Recalling notation from Theorem 2.2.3 that $k=\lfloor\rho/\eta\rfloor$, it is clear that $k\eta = \lfloor\rho/\eta\rfloor\ \eta \le \lfloor\rho\rfloor$, implying that $A_{k\eta} \subseteq A_{\lfloor\rho\rfloor}$, and therefore that $a_{k\eta} \le a_{\lfloor\rho\rfloor}$. Hence,

$$\begin{aligned} b = 1-a_{\lfloor\rho\rfloor} &\le 1-a_{k\eta} \\ &\le (1-a)(1+2a)^{-k}, \text{ from Theorem 2.2.3,} \\ &\le (½)(½)^{k}, \text{ since } a\geq ½, \\ &= (½)^{k+1} \\ &= (½)^{\lfloor\log_2(n)\rfloor+1}, \text{ since } k = \lfloor\rho/\eta\rfloor = \lfloor\log_2 n\rfloor, \\ &< (½)^{\log_2(n)}, \text{ since } \log_2 n < \lfloor\log_2 n\rfloor+1 \text{ and } ½<1, \\ &= 1/[2^{\log_2(n)}] = 1/n. \end{aligned}$$

Hence $|B|/n = b < 1/n$, implying that $|B|<1$. That is, B contains no vertices, and is thus the empty set. It follows immediately that $A_{\lfloor\rho\rfloor}=V$, since $\emptyset=B=V-A_{\lfloor\rho\rfloor}$ and $A_{\lfloor\rho\rfloor} \subseteq V$.

For any $v\in V$, consider $\{v\}_{\lfloor\rho\rfloor}$, the set of all vertices no further than $\lfloor\rho\rfloor$ from v. We shall show by contradiction that our diameter must be no greater than twice $\lfloor\rho\rfloor$, which is the desired result.

Suppose $|\{v\}_{\lfloor\rho\rfloor}| < n/2$. Define $A=V-\{v\}_{\lfloor\rho\rfloor}$, the set of all vertices further from v than $\lfloor\rho\rfloor$. By the supposition, $|A|>n/2$. Then, by the work just performed, $A_{\lfloor\rho\rfloor}=V$, which says that all vertices in V are no further than $\lfloor\rho\rfloor$ from set A. Since $v\in V$, we have that $d(v,A) \leq \lfloor\rho\rfloor$, a direct contradiction to the definition of set A as the set of all vertices further from v than $\lfloor\rho\rfloor$. Hence, the supposition is false and we have that $|\{v\}_{\lfloor\rho\rfloor}| \geq n/2$.

Again by the above work, we may conclude that $V=\{\{v\}_{\lfloor\rho\rfloor}\}_{\lfloor\rho\rfloor}$, which is the set of all vertices less than or equal to distance $\lfloor\rho\rfloor$ away from $\{v\}_{\lfloor\rho\rfloor}$, which is the same as the set of all vertices less than or equal to distance $2\lfloor\rho\rfloor$ away from v.

That is, from any $v\in V$, we can reach any other vertex in V within two steps of size $\lfloor\rho\rfloor$. By definition, we thus have that

$$\mathrm{Diam}(G) \leq 2\cdot\lfloor\rho\rfloor = 2\cdot\lfloor((2d/\lambda_1)^{1/2})\cdot\log_2 n\rfloor. \qquad \square$$

2.3 Chung's Upper Bound on Diameter

This section addresses Chung's [Ch] efforts to establish an upper bound on diameter. For the particular case of our Z_n-based Cayley graphs, her formula appears not to be an improvement over the bound established in Theorem 2.2.4 above (see Chapter 2.5). Hence, we present only the results of her work. First, we require the following definition:

DEFINITION 2.3.1: The **spectral radius** μ of a connected k-regular graph G is defined by

$$\mu = \mathit{max}\{|\xi|: \xi \text{ is an eigenvalue of } A(G) \text{ and } |\xi|<k\}. \qquad \square$$

That is, μ is the absolute value of the "largest" eigenvalue of A(G) less than k, where A(G) is the standard adjacency matrix of G.

One property of the standard adjacency matrix is that powers of the matrix result in entries which indicate the number of distinct walks between the vertices corresponding to the indices of that entry. Any power for which it has all nonzero entries thus corresponds to an upper bound for the diameter of the graph. Chung [Ch] has worked with this property to establish the following upper bound on the diameter of G:

THEOREM 2.3.2 ([Ch], Theorem 1): For a k-regular (connected) graph G on n vertices with spectral radius μ of the adjacency matrix A(G), we have

$$\mathrm{Diam}(G) \leq \lceil \log(n-1)/\log(k/\mu) \rceil. \qquad \square$$

2.4 The Relationship of λ_1 and μ

Recall from Chapter 2.2 that Q=K-A(G). In the case where G is k-regular, we have that K=kI, since K is a diagonal matrix with $K_{ii}=\deg(v_i)=k$. Any eigenvector f of Q is also an eigenvector of A(G), by

$$\lambda_i f = Qf = (K-A(G))f = Kf-A(G)f = kf-A(G)f$$

so

$$A(G)f = kf-\lambda_i f = (k-\lambda_i)f.$$

From our work in Chapter 4 we have that the eigenvalues ξ_i of A(G) are constrained as

$$|\xi_n| \le |\xi_{n-1}| \le \ldots \le |\xi_3| \le |\xi_2| \le \xi_1 = k.$$

Since we have k as an upper bound on $|\xi_i|$, then $|k-\lambda_i| \le k$, forcing $0 \le \lambda_i \le 2k$. From our work in Chapter 2.2, we know the lower bound on the eigenvalues of Q, so we now have that

$$0 = \lambda_0 < \lambda_1 = \lambda_1(G) \le \lambda_2 \le \ldots \le \lambda_{n-1} \le 2k.$$

Hence,

$$\xi_1 = k-\lambda_0 = k-0 = k;$$

$$|\xi_2| = \mathit{max}\{|k-\lambda_1|, |k-\lambda_{n-1}|\};$$

$$|\xi_3| = 2^{nd}\ \mathit{max}\{|k-\lambda_1|, |k-\lambda_{n-1}|, |k-\lambda_2|, |k-\lambda_{n-2}|\};$$

etc.

The first such $|\xi_i|$ that is strictly less than k is selected as the value for μ, the spectral radius. The net result of this is that

$$\mu = |k-\lambda_i| \text{ for some i such that } |k-\lambda_i| \geq |k-\lambda_1| \text{ since } \lambda_1>0.$$

Then

$$\mu \geq |k-\lambda_1| = k-\lambda_1,$$

and so

$$\mu \geq k-\lambda_1 \text{ or, equivalently, } \lambda_1 \geq k-\mu.$$

Note that, again from work in Chapter 4, if G is bipartite, then the multiplicity of ξ_i and $-\xi_i$ are the same for each i, thus forcing that $\lambda_1 = 2k-\lambda_{n-2}$, and hence equality holds for $\mu=k-\lambda_1$.

2.5 How Useful Are These Bounds?

Now that such upper bounds have been found, the question remains as to how good they are. Chung claims to have found a superior boundary. Though there may be some basis for her claim, we show that it is not true in the case of Cayley graphs based on groups of the form Z_n. In addition, we demonstrate that both are rather poor bounds when it comes to predicting the diameter of Cayley graphs on Z_n.

When studying the theory of Cayley graph structure, it is desirable to work with as general an underlying group as is possible. Since λ_1 is often difficult to determine in general for a large group, we initially used the two diameter bound formulas (Theorems 2.2.4 and 2.3.2) to examine the constraints they provide for λ_1 with a group of known diameter. As such a model, we used Z_{21} with the generating set of $S=\{3,7\}$. This Cayley graph is small enough to draw and find by exhaustion that the diameter is 4. Then we have from Theorem 2.2.4 [AM] that

$$\begin{aligned} & \mathrm{Diam}(G) \le 2\cdot\lfloor(\log_2 n)(2d/\lambda_1)^{½}\rfloor \\ \Rightarrow\quad & \mathrm{Diam}(G) \le 2(\log_2 n)(2d/\lambda_1)^{½} \\ \Rightarrow\quad & \mathrm{Diam}(G)^2 \le 4(\log_2 n)^2(2d/\lambda_1) \end{aligned}$$

$$\Rightarrow \quad \lambda_1 \le 8d(\log_2 n)^2/\mathrm{Diam}(G)^2$$

$$\Rightarrow \quad \lambda_1 \le 8\cdot 4\cdot(\log_2 21)^2/4^2$$

$$\Rightarrow \quad \lambda_1 \le 2\cdot(4.39)^2 \approx 38.6.$$

This is clearly not a helpful bound since we already know from Chapter 2.4 that $\lambda_1 \le 2k = 8$. Similarly, we have from Theorem 2.3.2 that

$$\mathrm{Diam}(G) \le \lceil \log(n-1)/\log(k/\mu) \rceil$$

$$\Rightarrow \quad \mathrm{Diam}(G)-1 \le \log(n-1)/\log(k/\mu)$$

$$\Rightarrow \quad 3 \le \log(20)/\log(k/\mu)$$

$$\Rightarrow \quad \log(k/\mu) \le \log(20)/3 \approx .4337$$

$$\Rightarrow \quad 4/\mu \le 10^{.4337} \approx 2.714$$

$$\Rightarrow \quad \mu \ge 4/(2.714) = 1.474.$$

But we have that $\mu \ge |k-\lambda_1|$, so even if we make the assumptions that $|k-\lambda_1| = \mu \ge 1.474$, we may only conclude that

$$4-1.474 \approx 2.53 \ge \lambda_1 \quad \text{or} \quad 4-(-1.474) = 5.474 \le \lambda_1$$

which doesn't provide us with any useful bounds on λ_1 at all. Certainly, evaluating their usefulness in this "reverse engineering" direction with only one example is not definitive, but Theorem 2.2.3 appears to be woeful while Theorem 2.3.2 doesn't help at all.

However, working in the opposite direction, we can rather easily evaluate the two upper bounds on the diameters of Cayley graphs of the cyclic groups Z_n by using the Lovasz algorithm discussed in Appendix D to find exact values of λ_1. In Chapter 6, we have developed a method of finding an algorithmic upper bound on the diameters of such graphs, and below in Table 2.5.1 we compare those diameter bounds with those obtained from Theorems 2.2.4 and 2.3.2.

TABLE 2.5.1: COMPARISONS OF UPPER BOUNDS ON DIAMETERS OF CAYLEY GRAPHS OF VARIOUS Z_n's FOUND FROM CHAPTER 6 METHODS AND THEOREMS 2.2.4 AND 2.3.2

Group, Generating Set, and Degree	Alg. Diam: Ch 6	λ_1: Lovasz	Diam: Thm. 2.2.4 [AM]	μ: Lovasz	Diam: Thm. 2.3.2 [Ch]
Z_{21}, S={1,4}, k=4	4	1.358	20	3.444	20
Z_{32}, S={1,5}, k=4	5	0.927*	28	3.072*	14
Z_{45}, S={1, 6}, k=4	6	0.681	36	3.722	53
Z_{100}, S={1, 10}, k=4	10	0.382	60	3.902	186
Z_{1024}, S={1, 32}, k=4	32	0.038	280	3.990	2866
Z_{1024}, S={1,10,100}, k=6	14	0.369	114	5.818	225
Z_{1024}, S={1,6,36,216}, k=8	9	1.201	72	7.142	62
Z_{1024}, S={1,4,16,32,256}, k=10	8	2.000	30	8.304	38

* note that $\lambda_1+\mu=4$ in this case. That is, $k-\lambda_1=\mu$, so that we get a diameter from Theorem 2.3.2 that is lower than usual when considering the other results.

The clear result of all these examples is that the upper bounds found by the two theorems shown in this chapter are so soft as to be useless, at least for the case of the Cayley graphs on Z_n. In addition, Chung's claim that Theorem 2.3.2 is superior to Theorem 2.2.4 does not seem valid for such Cayley graphs. (In fact, it appears to be generally worse for the cases examined.) It should also be noted that Theorem 2.2.4 is a stronger theorem in that it does not require the graph to be k-regular as does Theorem 2.3.2. Since all Cayley graphs are by definition regular, this is not significant to our findings, but it does indicate that Theorem 2.2.4 is applicable over a much broader set of graphs. In theory, adapting it to cover just the k-regular graphs could improve its upper bound, thereby making it far better than the bounds of Theorem 2.3.2 established by Chung.

Upon reflection, it is clear that Theorem 2.3.2 is quite sensitive to small variations in μ whenever μ is quite close to k in value. In this instance, the value of $\log(k/\mu)$ can skyrocket, resulting in a poor upper bound. (Note that the diameter upper bound for Z_{1024} when $S=\{1,32\}$ is 2866!) On the other hand, Theorem 2.2.4 is relatively insensitive to such variations in λ_1 since it follows the square root of $2d/\lambda_1$. In those occasional cases when $k-\lambda_1=\mu$, the value of μ is as far away as possible from k, and Theorem 2.3.2 may provide a somewhat better upper bound.

(See row two of Table 2.5.1 for such a case.) Hence, the wide fluctuations of the bounds supplied by Theorem 2.3.2.

CHAPTER 3

KAZHDAN CONSTANTS AS BOUNDS ON VARIOUS CONSTANTS

3.1 Introduction

Constructing a processor network model with a relatively small diameter is desirable since it improves the chances of low travel time for information through that network. However, as demonstrated above, it may not always be possible to establish good bounds on the diameter of various classes of graphs.

Thus, much of the theory under current consideration has been focused on alternative characteristics of a graph. Two of these characteristics are: the expanding constant, related to the rate at which a subset of vertices connects to neighboring vertices; and the Cheeger constant, a similar constant using edges instead of vertices.

This chapter follows Bacher and de la Harpe [BdlH] in their development of Kazhdan constants and their use as bounds on the expanding constant, the Cheeger constant, and λ_1 for finite graphs.

3.2 The Expanding Constant and the Cheeger Constant

Here we define the expanding constant and the Cheeger constant and then follow the proof of a theorem which establishes relationships among both constants and λ_1.

Let $G=(V,E)$ be a finite simple (i.e., no loops and no multiple edges between pairs of vertices) graph on vertex set V and edge set E. Let A and B partition V. (That is, $A\subset V$ and $B=V-A$.) Note that this implies that $\rho=1$. Then we have the following definitions.

DEFINITION 3.2.1: The **boundary** of subset A is

$$\partial A = \{v\in B\text{: } v \text{ is adjacent to a vertex in } A\}. \qquad \square$$

Hence, ∂A is, in essence, all the immediate "neighbors" of set A. Note that ∂A does not contain any of the vertices in set A itself, so we are counting only "new neighbors" of the vertices in set A.

DEFINITION 3.2.2: The **bridge** of sets A and B is

$E(A,B) = \{e \in E: e=(v_a, v_b) \text{ where } v_a \in A \text{ and } v_b \in B\}$. □

Thus, the bridge is the set of all edges "bridging the gap" between set A and set B.

DEFINITION 3.2.3: The **expanding constant** of G is

$c_{max} = max\{c > 0: |\partial A| \geq c(1-|A|/n)|A| \text{ for every } A \subset V,\ A \neq \emptyset\}$,

or, alternatively,

$c_{max} = max\{c>0: |\partial A|/|A| \geq c|B|/n \text{ for every } A \subset V,\ A \neq \emptyset\}$. □

Thus, to find the expanding constant, consider all possible subsets $\emptyset \neq A_i \subset V$. Find c_i for each such subset where

$$c_i = (n/|B|)(|\partial A|/|A|).$$

The minimum such c_i will equal c_{max} since it will be the maximum value of c that still keeps the requirements true. Hence c_{max} establishes the minimum fraction A will grow when absorbing immediate neighbors relative to the fraction of vertices not already inside A.

DEFINITION 3.2.4: The **Cheeger constant** of G is

$$h(G) = \mathit{min}\{|E(A,B)|/\mathit{min}\{|A|,|B|\}: A,B \text{ partition } V\}. \quad \square$$

Thus, the Cheeger constant is the smallest ratio of "bridge edges" to the size of the smaller subset of V, for all possible partitions A,B of V. For a finite graph, h(G) may be tediously evaluated by testing the edge connectivity of every possible subset of V from order 1 to order n/2, and then dividing by the order of that subset. The smallest such quotient will yield h(G). As an immediate consequence of the definition we know that d_{min}, the minimum degree of any one vertex in V, is an upper bound for h(G), since we have for every v that deg(v) = |E(v,V-v)/|{v}|. In addition, if G is not connected, then h(G)=0 since there would clearly be a subset with no bridge edges to connect it to the rest of the vertices. Then it is possible to link the values of these constants to each other and to λ_1 in the following ways.

THEOREM 3.2.5 ([BdlH], Appendix, Proposition 5): Let G be a connected graph of maximum degree d. Let λ_1 be the smallest nonzero eigenvalue of Q as defined in Chapter 2.2.

Then

(a) $c_{max} \geq h(G)/d$

(b) $h(G) \geq \frac{1}{2}c_{max}$

(c) $h(G) \geq \frac{1}{2}\lambda_1$

(d) $\lambda_1 \geq h(G)^2/(2d)$.

Proof:

(a) By the definition of c_{max}, there exists a subset $A \subset V$ such that

$$|\partial A|/|A| = c_{max}(1-|A|/n) = c_{max}(|B|/n).$$

Hence,

$$c_{max} = [|\partial A|/|A|](n/|B|) = |\partial A| n(|A||B|)^{-1},$$

and so

$$c_{max}d = d|\partial A| n(|A||B|)^{-1}.$$

Clearly, the maximum possible number of bridge edges between sets A and B can occur only when each vertex in B that is adjacent (an immediate neighbor) to set A is, in fact, adjacent to d vertices in set A. Otherwise, the number of bridge edges is certainly less than $d|\partial A|$. Thus, we have that

$$d|\partial A| \geq |E(A,B)|,$$

and hence

$$c_{max}d \geq n|E(A,B)|/(|A||B|).$$

Without loss of generality, assume that $|A| \leq |B|$. Then

$n/|B| \geq 1$ and $|E(A,B)|/|A| \geq h(G)$, so that

$$c_{max} d \geq 1 \cdot h(G) = h(G),$$

thus yielding the desired result that

$$c_{max} \geq h(G)/d.$$

(b) By the definition of $h(G)$, there exists a partition A,B of V with $|A| \leq |B|$ such that $|E(A,B)|/\mathit{min}\{|A|,|B|\} = h(G)$. Then

$$\begin{aligned} h(G) &= |E(A,B)|/|A| \\ &\geq |\partial A|/|A|, \text{ since } \leq \text{ one new neighbor per bridge edge,} \\ &\geq c_{max}(1-|A|/n), \text{ by definition,} \\ &\geq \tfrac{1}{2} c_{max}, \text{ since } |A| \leq n/2. \end{aligned}$$

(c) From Theorem 2.2.1, we have

$$\lambda_1 n \leq \rho^{-2}(n/|A| + n/|B|)(|E|-|E_A|-|E_B|),$$

where A and B are two non-empty disjoint subsets of V, ρ is the distance between set A and set B, E is the edge set of G, and E_A and E_B are the edges internal to sets A and B, respectively. Then, for any partition of V, $\rho=1$ and $|E|-|E_A|-|E_B| = |E(A,B)|$ will be the bridge edge set. Thus, we have that

$$\begin{aligned}\lambda_1 n &\leq \rho^{-2}(n/|A| + n/|B|)(|E|-|E_A|-|E_B|)\\ &\leq 1\cdot(n/|A| + n/|B|)\cdot|E(A,B)|,\end{aligned}$$

and so

$$\begin{aligned}\lambda_1 &\leq (1/|A| + 1/|B|)|E(A,B)|\\ &= [(|A|+|B|)/(|A||B|)]\cdot|E(A,B)|, \text{ for partition A,B of V.}\end{aligned}$$

As in part (b) above, choose a partition A,B of V such that $h(G) = |E(A,B)|/min\{|A|,|B|\}$ and, without loss of generality, that $|A|\leq|B|$. Then

$$\lambda_1 \leq [(|A|+|B|)/(|A||B|)]\cdot|E(A,B)|$$

$$= [(|A|+|B|)/|B|]\cdot[|E(A,B)|/|A|]$$

$$= [(|A|+|B|)/|B|]\cdot h(G)$$

$$\leq [(|B|+|B|)/|B|]\cdot h(G), \text{ since } |A|\leq|B|,$$

$$= 2h(G).$$

Hence,

$$h(G) \geq \tfrac{1}{2}\lambda_1.$$

(d) (Per [Lub], Proposition 4.2.4.) (Note that initial attempts to follow [BMS] Theorem 3.2 for an equivalent proof turned up an error between lines 6 and 7 on page 206. In addition, [BMS] uses the isoperimetric number, which is not as strong a conclusion.) Let g be a unit eigenvector corresponding to λ_1 of Q. That is, let $Qg=\lambda_1 g$ and $(g,g)=1$. Then

$$(Cg,Cg) = (Qg,g) = (\lambda_1 g,g) = \lambda_1(g,g).$$

Let $V^+ = \{v\in V:\ g(v)>0\}$ and define the related n-tuple f as

$$f(v) = \begin{cases} g(v) & \text{if } v\in V^+ \\ 0 & \text{otherwise.} \end{cases}$$

Since we could replace g with -g, we may assume without loss of generality that $|V^+| \leq \frac{1}{2}|V|$. Then we have that

$$\begin{aligned}(Cf,Cf) &= (Qf,f)\\ &= (f,Qf)\\ &= \sum_{v\in V} [f(v) \sum_{u\in V} Q_{vu}(f(v)-f(u))],\end{aligned}$$

since the i^{th} entry of f (the i^{th} term in the first summation) is to be multiplied by the i^{th} entry of Qf, which is simply the dot product of the i^{th} row of Q with f, as represented by the second summation.

However, we have $g(v)=f(v)$ for every $v\in V^+$, so the right-hand side may, by working with the summations, be written as

$$\begin{aligned}&= \sum_{v\in V^+} [g(v) \sum_{u\in V} Q_{vu}(g(v)-f(u))]\\ &= \sum_{v\in V^+} [g(v)\{\sum_{u\in V^+} Q_{vu}(g(v)-f(u)) + \sum_{u\notin V^+} Q_{vu}(g(v)-f(u))\}]\\ &= \sum_{v\in V^+} [g(v)\{\sum_{u\in V^+} Q_{vu}(g(v)-g(u)) + \sum_{u\notin V^+} Q_{vu}(g(v)-f(u))\}]\\ &= \sum_{v\in V^+} [g(v)(\sum_{u\in V^+} Q_{vu}(g(v)-g(u)))]\\ &\qquad\qquad + \sum_{v\in V^+} [g(v)(\sum_{u\notin V^+} Q_{vu}(g(v)-f(u)))]\end{aligned}$$

$$= \sum_{v\in V^+} [g(v)(\sum_{u\in V^+} Q_{vu}(g(v)-g(u)))] + \sum_{v\in V^+} [g(v)(\sum_{u\notin V^+} Q_{vu}g(v))],$$

since $f(u)=0$ for $u\notin V^+$.

Recalling that $g(v)$ is positive for every $v\in V^+$, that Q_{vu} is either zero or 1 for each entry, and that $g(v)$ is either zero or negative for every $v\notin V^+$, we have for the right-hand term

$$\sum_{v\in V^+} [g(v)(\sum_{u\notin V^+} Q_{vu}g(v))] \leq 0.$$

Hence

$$(Cf,Cf) \leq \sum_{v\in V^+} [g(v)(\sum_{u\in V^+} Q_{vu}(g(v)-g(u)))].$$

But $\sum_{u\in V^+} Q_{vu}(g(v)-g(u))$ is simply the v^{th} entry of the n-tuple Qg, so we have that

$$\begin{aligned}\sum_{v\in V^+} [g(v)(\sum_{u\in V^+} Q_{vu}(g(v)-g(u)))] &= \sum_{v\in V^+} g(v)Qg(v)\\ &= \sum_{v\in V^+} g(v)\lambda_1 g(v)\\ &= \lambda_1 \sum_{v\in V^+} g(v)^2\\ &= \lambda_1(f,f).\end{aligned}$$

Thus, we have that

$$(Cf,Cf) \leq \lambda_1(f,f).$$

Let α be the constant defined by $\alpha = \sum_{e\in E} |f^2(e^+)-f^2(e^-)|$. Then

$$\begin{aligned}\alpha &= \sum_{e\in E} |f^2(e^+)-f^2(e^-)| \\ &= \sum_{e\in E} |f(e^+)+f(e^-)|\cdot|f(e^+)-f(e^-)| \\ &= \{[\sum_{e\in E} |f(e^+)+f(e^-)|\cdot|f(e^+)-f(e^-)|]^2\}^{½}.\end{aligned}$$

Claim: $$[\sum_{e\in E} |f(e^+)+f(e^-)|\cdot|f(e^+)-f(e^-)|]^2 \leq \sum_{e\in E} |f(e^+)+f(e^-)|^2\cdot\sum_{e\in E} |f(e^+)-f(e^-)|^2.$$

Let $a_i = |f(e^+)+f(e^-)|$ and $b_i = |f(e^+)-f(e^-)|$ for edge e_i.

Suppose $|E|=1$.

Then $(a_1b_1)^2 = a_1^2b_1^2$, so the claim is true when $|E|=1$.

Let $|E|=n$ be a value for which the claim is true.

Then

$$\begin{aligned}&(a_1b_1+...+a_nb_n + a_{n+1}b_{n+1})^2 \\ &\quad = [(a_1b_1+...+a_nb_n) + a_{n+1}b_{n+1}]^2 \\ &\quad = (a_1b_1+...+a_nb_n)^2 + 2(a_1b_1+...+a_nb_n)(a_{n+1}b_{n+1}) + (a_{n+1}b_{n+1})^2,\end{aligned}$$

while

$$\begin{aligned}&(a_1^2+...+a_{n+1}^2)(b_1^2+...+b_{n+1}^2) \\ &\quad = [(a_1^2+...+a_n^2) + a_{n+1}^2][(b_1^2+...+b_n^2) + b_{n+1}^2]\end{aligned}$$

$$= (a_1^2+\ldots+a_n^2)(b_1^2+\ldots+b_n^2) + (a_1^2+\ldots+a_n^2)(b_{n+1})^2$$
$$+ (b_1^2+\ldots+b_n^2)(a_{n+1})^2 + (a_{n+1}b_{n+1})^2.$$

By the inductive hypothesis we have

$$(a_1b_1+\ldots+a_nb_n)^2 \le (a_1^2+\ldots+a_n^2)(b_1^2+\ldots+b_n^2).$$

Hence, it remains to compare the middle terms. That is

$$2(a_1b_1+ \ldots +a_nb_n)(a_{n+1}b_{n+1})$$
$$\le (a_1^2+ \ldots +a_n^2)(b_{n+1})^2+(b_1^2+ \ldots+b_n^2)(a_{n+1})^2$$
$$\Leftrightarrow \quad 2a_1b_1a_{n+1}b_{n+1}+\ldots+2a_nb_na_{n+1}b_{n+1}$$
$$\le a_1^2b_{n+1}^2 +\ldots+ a_n^2b_{n+1}^2 +\ldots+ b_1^2a_{n+1}^2 +\ldots+ b_n^2a_{n+1}^2$$
$$\Leftrightarrow \quad 0 \le a_1^2b_{n+1}^2 - 2a_1b_1a_{n+1}b_{n+1} + b_1^2a_{n+1}^2 +\ldots+ a_n^2b_{n+1}^2$$
$$- 2a_nb_na_{n+1}b_{n+1} + b_n^2a_{n+1}^2$$
$$\Leftrightarrow \quad 0 \le (a_1b_{n+1} - b_1a_{n+1})^2 +\ldots+ (a_nb_{n+1} - b_na_{n+1})^2,$$

which, of course, is true since the sum of the squares of real numbers is never negative.

Thus the claim is true, and we have that

$$\begin{aligned}\alpha &= \{[\sum_{e\in E} |f(e^+)+f(e^-)|\cdot|f(e^+)-f(e^-)|]^2\}^{1/2} \\ &\leq \{\sum_{e\in E} |f(e^+)+f(e^-)|^2\cdot\sum_{e\in E} |f(e^+)-f(e^-)|^2\}^{1/2}.\end{aligned}$$

By completing the square, we observe that

$$\sum_{e\in E} |f(e^+)+f(e^-)|^2 \leq 2\sum_{e\in E} [f(e^+)^2+f(e^-)^2].$$

Then, since $\sum_{e\in E} |f(e^+)-f(e^-)|^2 = (Cf,Cf)$, by definition, we have by substitution that

$$\alpha \leq \{2\sum_{e\in E} [f^2(e^+)+f^2(e^-)]\}^{1/2}(Cf,Cf)^{1/2}.$$

Since each vertex in G appears in $\sum_{e\in E} |f^2(e^+)+f^2(e^-)|$ precisely as many times as the number of edges adjacent to it (i.e., its degree), no vertex appears more than d times, where d is the maximum degree of G. Therefore

$$\begin{aligned}\alpha &\leq \{2\sum_{e\in E} [f^2(e^+)+f^2(e^-)]\}^{1/2}(Cf,Cf)^{1/2} \\ &\leq \{2[d\sum_{v\in V} f^2(v)]\}^{1/2}(Cf,Cf)^{1/2} \\ &= (2d)^{1/2}(\sum_{v\in V} f^2(v))^{1/2}(Cf,Cf)^{1/2}\end{aligned}$$

$$= (2d)^{½}(f,f)^{½}(Cf,Cf)^{½}.$$

Since we showed above that $(Cf,Cf) \le \lambda_1(f,f)$, we have that $(Cf,Cf)^{½} \le (\lambda_1)^{½}(f,f)^{½}$ and thus

$$\alpha \le (2d\lambda_1)^{½}(f,f) \qquad (*)$$

by substitution.

It remains to be shown that $\alpha \ge h(G)(f,f)$.

Recall that $\alpha = \sum_{e\in E} |f^2(e^+)-f^2(e^-)|$. Note that in evaluating α by considering each $e\in E$, any edge $e=(v,u)$ where $f(v)=f(u)$ contributes nothing to the summation.

Let $V=\{v_1,\ldots,v_n\}$ denote the vertex set of G and let $f(v_i)$ denote the i^{th} entry in the n-tuple f. There are n such entries, some of which may be equal. Thus, we may specify a strictly increasing sequence of n real numbers

$$0=y_0 < y_1 < y_2 < \ldots < y_n$$

where each of the real number entries in f appears once in the sequence. For each entry of f that is a repeated value, the sequence contains a "filler" number of some intermediate value in order to yield the full sequence of n real numbers. For example, perhaps $f(v_i) = f(v_j) = y_s$. Then, at least one value y_q will not be equal to any of the entries in the n-tuple f, since there are at most n-2 distinct values of entries left. Y_q thus becomes a "filler" number in the sequence, unassociated with any vertices in G.

Define F_k as the subgraph of G induced on the vertices

$$V_k = \{v: f(v) \geq y_k\}$$

so that V_k is the set of all the vertices associated with an entry of f that is greater than or equal to y_k.

Let D_k be the orientation of F_k whose edges are directed so that edge e goes from $e^{+}=v_i$ to $e^{+-}=v_j$ where $f(v_i) \geq f(v_j)$. This ensures that while evaluating a summation on any D_k, each term in the summation will be positive inside the absolute value sign, and therefore that it will be possible to eliminate them. That is, for each k

$$|f^2(e^+)-f^2(e^-)| = f^2(e^+)-f^2(e^-), \text{ for every } e \in D_k.$$

Define $S_k = \{e \in E(G):\ f(e^+)=y_k\}$, the set of all edges in G whose initial vertex in any D_k is assigned the value y_k in f. Then, it is possible to make the following statement about α:

$$\alpha = \sum_{e \in E} |f^2(e^+)-f^2(e^-)|$$

$$= \sum_{k=1}^{n} [\sum_{e \in S_k} (f^2(e^+)-f^2(e^-))],$$

since this double summation looks at edges whose initial vertex corresponds to an entry with value y_k in f, and looks at all possible k's (and thus all possible entries), it looks at each edge in E precisely once.

Claim: $$\sum_{k=1}^{n} [\sum_{e \in S_k} (f^2(e^+)-f^2(e^-))] = \sum_{k=1}^{n} [\sum_{e \in \partial V_k} (y_k^2-y_{k-1}^2)],$$

where $\partial V_k = \{e \in E:$ e has exactly one vertex in $V_k\}$. That is, ∂V_k is the "bridge edge" set of V_k.

To substantiate this claim, consider an edge e=(v,u) in

S_k. Its initial vertex v is such that $f(v)=y_k$ while its terminating vertex u has the property that $f(u)=y_q \le y_k$. If $f(u)=y_k$, then edge e contributes only zero to each summation. However, if $f(u)<y_k$, then e is clearly a bridge edge for each of the subgraphs $\{F_k, F_{k-1}, F_{k-2}, \ldots, F_{q+1}\}$, since vertex v is in each of the sets $\{V_k, V_{k-1}, V_{k-2}, \ldots, V_{q+1}\}$ yet vertex u is in none of them since $F(u)=y_q$. Hence, e appears in each of the bridge edge sets $\{\partial F_k, \partial F_{k-1}, \partial F_{k-2}, \ldots, \partial F_{q+1}\}$ and yet appears in no other bridge edge set since every other vertex set V_i either contains both v and u or contains neither v nor u. Hence, the edge e will contribute the terms

$$\{(y_k^2-y_{k-1}^2), (y_{k-1}^2-y_{k-2}^2), \ldots, (y_{q+1}^2-y_q^2)\}$$

to the right-hand side summation. But

$$\begin{aligned} y_k^2-y_q^2 &= y_k^2 - (y_{k-1}^2-y_{k-1}^2) - \ldots - (y_{q+1}^2-y_{q+1}^2) - y_q^2 \\ &= (y_k^2-y_{k-1}^2) + (y_{k-1}^2-y_{k-2}^2) + \ldots + (y_{q+1}^2-y_q^2), \end{aligned}$$

while

$$y_k^2-y_q^2 = f^2(e^+)-f^2(e^-)$$

which is precisely what each edge in S_k contributes to the left-hand side summation. Thus the claim is verified.

Then we have that

$$\alpha \ = \sum_{k=1}^{n} [\sum_{e\in\partial V_k} (y_k^2 - y_{k-1}^2)] \ .$$

For each value of k, each edge $e\in\partial V_k$ contributes the same value $(y_k^2 - y_{k-1}^2)$ to the summation. Hence, for each value of k, we get $|\partial V_k|(y_k^2 - y_{k-1}^2)$ added to the summation, so that

$$\alpha \ = \sum_{k=1}^{n} [|\partial V_k|(y_k^2 - y_{k-1}^2)] \ .$$

Recall that by our choice of g and our construction of f we have $|\partial V_k| \le \frac{1}{2}|V|$. Hence, by the definition of the Cheeger's constant we have that $|\partial V_k| \ge h|V_k|$ for every k from 1 through n. Then, by substitution, we have

$$\alpha \ \ge h(G) \sum_{k=1}^{n} [|V_k|(y_k^2 - y_{k-1}^2)]$$

$$= h(G) \{ \sum_{k=1}^{n} y_k^2 |V_k| \ - \ \sum_{k=1}^{n} y_{k-1}^2 |V_k| \}$$

$$= h(G)\{\sum_{k=1}^{n} y_k^2|V_k| \; - \; \sum_{j=0}^{n-1} y_j^2|V_{j+1}|\}, \text{ by an index shift,}$$

$$= h(G)\{\sum_{k=1}^{n-1} y_k^2|V_k| \; + \; y_n^2|V_n| \; - \; \sum_{j=1}^{n-1} y_j^2|V_{j+1}| \; - \; y_0^2|V_j|\}$$

$$= h(G)\{\sum_{k=1}^{n-1} y_k^2|V_k| \; + \; y_n^2|V_n| \; - \; \sum_{j=1}^{n-1} y_j^2|V_{j+1}|\}, \text{ since } y_0=0,$$

$$= h(G)\{\sum_{k=1}^{n-1} y_k^2(|V_k|-|V_{k+1}|)\} \; + \; h(G)y_n^2|V_n|.$$

Since there are no vertices in V_{n+1}, we have that $|V_{n+1}|=0$ and thus $h(G)y_n^2|V_{n+1}| = 0$. Hence, we can add zero to both sides as

$$\alpha \; = h(G)\{\sum_{k=1}^{n-1} y_k^2(|V_k|-|V_{k+1}|)\} \; + \; h(G)y_n^2|V_n| \; - \; h(G)y_n^2|V_{n+1}|,$$

$$= h(G)\sum_{k=1}^{n} y_k^2(|V_k|-|V_{k+1}|) .$$

Recall that V_k consists of all vertices whose corresponding entries in f have a value of y_k or greater. Similarly, V_{k+1} contains all the vertices whose corresponding entries in f have a value of y_{k+1} or greater. Thus, $(|V_k|-|V_{k+1}|)$ is simply the number of entries in f whose value is y_k

since the sequence was constructed in such a way that every vertex's corresponding entry in f has a value that is precisely one of the y_i's. So, as the above summation is evaluated at each value of k, we get y_k^2 times the number of vertices whose entries in f correspond to the value y_k. Then, for any value of k such that no vertex has an entry in f equal to y_k, $|V_k|-|V_{k+1}| = 0$, and that term contributes nothing to the sum. (Note that $|V_n|-|V_{n+1}| = |V_n|$ is the number of vertices whose entries in f have the value y_n.) Hence, working through the summation, we get that

$$\sum_{k=1}^{n} y_k^2(|V_k|-|V_{k+1}|) = \sum_{k=1}^{n} f(v_k)^2 = (f,f),$$

and so

$$\alpha \geq h(G)\cdot(f,f). \qquad (**)$$

Combining equation (*) above with equation (**) yields

$$(2d\lambda_1)^{½}(f,f) \geq \alpha \geq h(G)\cdot(f,f)$$

and so

$$(2d\lambda_1)^{½} \geq h(G)$$

or, equivalently

$$\lambda_1 \geq h^2(G)/(2d). \qquad \square$$

3.3 Background on Representations

This chapter briefly examines the subject of group representations as necessary background for a thorough understanding of the Kazhdan constants. Also presented are some special examples of such representations and several theorems particularly applicable to the evaluation of Kazhdan constants for abelian groups. Much of this material is from Diaconis' text [Dia], Chapter 2.

DEFINITION 3.3.1: The **representation** π of a group Γ is a group homomorphism which sends each element $g \in \Gamma$ to an invertible (hence square) matrix with complex entries. That is

$$\pi \colon \Gamma \mapsto GL(V)$$

where GL(V) is the set of linear maps (invertible matrices of size n×n, with dim(V)=n) on vector space V. □

The binary action in GL(V) is, of course, matrix multiplication, and each such matrix is denoted by $\pi(g)$. By the properties of homomorphisms, for $s,t \in \Gamma$ and e the identity of Γ, we have

(i) $\pi(st) = \pi(s)\pi(t)$

(ii) $\pi(e) = I_{n\times n}$

(iii) $\pi(e) = \pi(s^{-1}s) = \pi(s^{-1})\pi(s)$ so $\pi(s^{-1}) = [\pi(s)]^{-1}$.

Observe that the order of Γ and the size of the matrices need not be the same. For example, if $\mathrm{Ker}(\pi)$ contains other than the identity of Γ, then clearly $|\pi(\Gamma)|<|\Gamma|$, so that $\pi(\Gamma)$ may be represented by a set of $|\pi(\Gamma)|\times|\pi(\Gamma)|$ matrices. In addition, it may simply be possible to provide a group in GL(V) where $\dim(V)<|\Gamma|$ that is nonetheless isomorphic to the original group Γ. In any case, the **dimension of** π is designated to be the same as the dimension of the vector space V. In theory, the dim(V) could actually be larger than $|\Gamma|$, by adding extra ones to the diagonal entries of additional (unused) dimensions. However, in practice this seems unnecessary, as the permutation matrices (also known as the regular representation of Γ) are the most commonly used representation with matrices of size $|\Gamma|$.

DEFINITION 3.3.2: A representation π has a **subrepresentation** π_w (restricted to W) if W is a subspace of V that is stable under Γ. That is, if $\pi(g)w \in W$ for every $g \in \Gamma$ and for every $w \in W$. A representation π is **irreducible** if it has no non-trivial subrepresentations. □

Then a representation π is irreducible if there is no subspace -- except for V itself and the zero subspace -- which is stable under Γ. In essence, irreducibility corresponds to saying that you have used the minimum size possible for the matrices used to form the group representation. Otherwise, if it could have been represented by a group of matrices of smaller size, then a suitable change of basis matrix applied to each of the original representations' matrices would have allowed at least one dimension to go unused. Hence, a representation is irreducible if the dimension of π is the minimum possible matrix size in any group of matrices that is isomorphic to $\pi(\Gamma)$.

Presented here are some examples of representations of S_n, the permutation group on n letters:

a) The trivial representation. That is, π sends each $\rho \epsilon S_n$ to the identity matrix. Thus, we have $Ker(\pi)=S_n$. Now, the dimension of π could be anything, since we could choose to send each ρ to any size matrix I, and still satisfy all the homomorphism properties, since $\pi(S_n)$ is a trivial group. However, any size that is chosen other than matrices of size 1×1 will be reducible, since clearly any subspace W of space

V is stable under S_n when $\pi(\rho)w = Iw = w$ for every $w \in W$. Thus, it is conventional to consider this trivial representation as the homomorphism sending each $\rho \in S_n$ to 1, where 1 may be viewed as a 1×1 matrix. The dimension of π in this case is thus 1.

b) The alternating representation. That is, π sends a permutation $\rho \in S_n$ to -1 or +1, depending on whether ρ is an odd or even permutation, respectively. (Observe that this could again be any dimension representation, using -I and +I of the desired size as the target matrices.) However, as for the trivial representation, convention dictates that this representation has dimension 1, since we are sending to 1×1 matrices.

c) The classic regular (or permutation) representation. Here, π sends a permutation $\rho \in S_n$ to its so-called permutation matrix M_ρ. Consider how ρ acts on the n elements h_1, $h_2, \ldots,$ h_n. For each i from 1 through n, ρ sends h_i to some other element $\rho(h_i)=h_j$ in the set. Then M_ρ is the invertible n×n matrix in which the elements of the i^{th} column (for each i) are all zero except for the entry in the j^{th} row, which is a 1. As such, multiplication of an n-tuple by any such matrix $\pi(\rho)$ simply permutes the entries of the n-tuple in a manner corresponding to the way the n letters are shuffled by the permutation ρ. Clearly, π is n-dimensional. Note that it is simple to show that π is reducible. Consider the set of all constant real-valued n-tuples, which is a 1-dimensional subspace $W \subset \mathbb{R}^n$. Obviously, the shuffling of the entries of a constant n-tuple yields the same n-tuple, so W is stable under the subgroup of permutations Γ_ρ as represented by π. The complement of W is the subspace $W^0 = \{f \in \mathbb{R}^n \colon \sum f_i = 0\}$, since any vector in $\mathbb{R}^n$ may be formed from these two subspaces, yet the only constant n-tuple whose entries sum to zero is the zero vector itself. Observing that the sum of the entries of an n-tuple does not change merely from shuffling their order of addition, it is clear that W^0 is also stable under Γ_ρ as represented by π. Recall that any group $\Gamma = \{h_1, \ldots, h_n\}$ is isomorphic to its permutation group Γ_ρ by $\varphi\colon \Gamma \mapsto \Gamma_\rho$, where φ is defined by

$\varphi(h)=\rho_h \in \Gamma_\rho$, and $\rho_h(h_i) = h*h_i \in \Gamma$ for every i. Hence, we may construct the regular representation of any group.

The fact that both W and its complement W^0 are stable in the above example gives rise to the following theorem:

THEOREM 3.3.3 ([Dia], Ch. 2, Thm. 1): Let $\pi\colon \Gamma \mapsto GL(V)$ be a linear representation in V and let W be a subspace of V stable under Γ. Then there is a complement W^0 (where $V=W+W^0$ and $W\cap W^0=0$) that is stable under Γ.

Proof: Let $\alpha=\{\alpha_1,\ldots,\alpha_s\}$ be a basis for W and $\beta=\{\beta_1,\ldots,\beta_t\}$ be a basis for W^0.

Suppose there exists $x\in W^0$ such that $\pi(g)x \notin W^0$ for some $g\in\Gamma$. Then

$$\begin{aligned}\pi(g)x &= a_1\alpha_1+\ldots+a_s\alpha_s+b_1\beta_1+\ldots+b_t\beta_t, \text{ where at least one } a_i\neq 0,\\ &= w + w^0, \text{ for some nonzero } w\in W \text{ and some } w^0\in W^0.\end{aligned}$$

But

$$\begin{aligned} x = Ix &= [\pi(g^{-1})\pi(g)]x = \pi(g^{-1})\,[\pi(g)x] \\ &= \pi(g^{-1})\,[w+w^0] \\ &= \pi(g^{-1})w + \pi(g^{-1})w^0. \end{aligned}$$

However, since W is stable under Γ, we have $\pi(g^{-1})w \in W$. Also, $\pi(g^{-1})w \neq 0$ since if $Aw=0$, then $w = A^{-1}Aw = A^{-1}0 = 0$, contradicting that w is nonzero. But this says that x is composed partly of a nonzero vector in W, a contradiction to $x \in W^0$. Hence, the supposition that there exists such an $x \in W^0$ is false. That is, W^0 is stable under Γ by the representation of π. □

COROLLARY 3.3.4 ([Dia], Chapter 2, Theorem 2): Every representation is a direct sum of irreducible representations.

Proof: Clearly, by induction, Theorem 3.3.3 may be applied repeatedly, yielding a given representation π

decomposed (eventually) into irreducible parts. (The terminology is that the representation π on V splits into the **direct sum** of W and W^0, and we write $V = W \oplus W^0$.) □

The above theorem and its corollary are extremely important since they allow study of the action of Γ on V by separately studying the action on its irreducible subrepresentations. In fact, further exploration of representations gives rise to the following two theorems (given without proof here) which prove useful in understanding and evaluating Kazhdan constants.

THEOREM 3.3.5 ([Dia], Ch. 2, Coro. 1 to Prop. 5): Every irreducible representation W_i is contained in the regular representation with multiplicity equal to its degree. □

THEOREM 3.3.6 ([Dia], Ch. 2, Theorem 8): The following properties are equivalent:

(i) Γ is abelian.

(ii) All irreducible representations of Γ have degree 1 over the complex numbers. □

Finally, we present two more definitions needed to untangle the meaning of the Kazhdan constants. These are:

DEFINITION 3.3.7: A representation π of a group Γ is **unitary** if every matrix in the group representation is a unitary matrix. □

Definition 3.3.7 is somewhat reflexive in that it needs elaboration on the subject of unitary matrices. A matrix B is unitary if it is n×n and satisfies $BB^* = B^*B = I$, where B^* is the conjugate transpose of B. Typically, the term unitary is reserved for those times where B contains complex (non-real) entries, while such a matrix with exclusively real entries is called orthogonal. Thus, if π is a unitary representation on the group Γ, then $\|\pi(h)v\| = \|v\|$ for every $h\in\Gamma$ and $v\in V$, the vector space of dimension n. Clearly, if $V\subset\mathbb{R}^n$, then to say that π is unitary is to say that each matrix in the representation is "length preserving".

DEFINITION 3.3.8 ([AM], Definition 4.5): A unitary representation π of a (finite) group Γ into the vector space V is called **essentially nontrivial** if, for any nonzero vector $f\in V$, there exists an $h\in\Gamma$ such that $\pi(h)f \neq f$. □

The concept of essentially nontrivial may be viewed as saying that *every* nonzero n-tuple in V is "moved" by *at least one* of the $n\times n$ matrices in the representation π. (Here, "moved" is used in the sense that a different n-tuple emerges as the product.)

If π is essentially nontrivial then the isomorphism of Γ onto the group M_π cannot be represented by a group of smaller size matrices, but, in fact, must be represented by a group of n×n matrices. To justify this conclusion, suppose it was possible to represent Γ by a group of $(n-1)\times(n-1)$ matrices. Then it would also be possible to represent Γ with the same group of matrices "embedded" into the lower right-hand corner of n×n matrices with the only nonzero entry in row one and column one being a 1 in the upper left-hand corner. Call this representation φ and the embedded group of matrices M_φ. Then we would have constructed the isomorphisms $\pi\colon \Gamma \mapsto M_\pi$ and $\varphi\colon \Gamma \mapsto M_\varphi$ and thus the isomorphism

$$\Phi = \varphi\circ\pi^{-1}\colon M_\pi \mapsto M_\varphi.$$

But this is simply equivalent to a change of basis matrix. Since M_φ clearly has the n-tuple $f=(c,0,\ldots,0)$ as a simultaneous eigenvector, M_π must have as an eigenvector the n-tuple obtained from f after the same change of basis. But this contradicts that π is essentially non-trivial, so no such representation of Γ exists onto a group of $(n-1)\times(n-1)$ matrices. Of course, any smaller-sized matrices are also ruled out, since they may be embedded into an $(n-1)\times(n-1)$ format. Hence, there is no (1-dimensional) subspace of V for which π has a trivial subrepresentation, so π is nontrivial in an essential way.

3.4 The Kazhdan Constants

Let Γ be a countable group. We denote by Γ^* the set of all unitary representations of Γ in separable Hilbert spaces. We also denote by Γ'' the subset of Γ^* consisting of all irreducible representations. Let S be a finite generating set for Γ, and let $\pi\in\Gamma^*$ be a unitary representation of Γ on some Hilbert space $\mathfrak{H}_\pi$. Then we have the following definition of Kazhdan constants:

DEFINITION 3.4.1 ([BdlH]): The Kazhdan constants are

(a) $\kappa_\Gamma(S,\pi) = \mathit{inf}\{\mathit{max}\{\|\pi(s)\xi-\xi\|: s\in S\}: \xi\in\mathfrak{H}_\pi \text{ and } \|\xi\|=1\}$

(b) $\kappa_\Gamma(S) = \mathit{inf}\{\kappa_\Gamma(S,\pi): \pi$ is essentially nontrivial$\}$

(c) $\hat{\kappa}_\Gamma(S) = \mathit{inf}\{\kappa_\Gamma(S,\pi): \pi$ is irred. and ess. nontr.$\}$ □

In keeping with these definitions, we say that Γ **is Kazhdan** if $\kappa_\Gamma(S)>0$. We now have the tools to work through the following theorem.

THEOREM 3.4.2 ([BdlH], Appendix, Proposition 6): Let $G=G(V,E)$ be the Cayley graph of a finite group Γ of order n with respect to a set $S=S^{-1}$ of $|S|=k$ generators, with $1\notin S$ (i.e., no loops). Then

$$\text{(a)} \qquad c_{max} \geq \tfrac{1}{2}\kappa_\Gamma(S)^2$$

$$\text{(b)} \qquad \hat{\kappa}_\Gamma(S) \geq \lambda_1/k.$$

Proof:

(a) ([Lub], Prop 4.3.1): Clearly, if Γ is not Kazhdan, then $\kappa_\Gamma(S)=0$, so $c_{max} \geq \tfrac{1}{2}\kappa_\Gamma(S)^2 = 0$ since $c_{max}>0$ by definition. If Γ is Kazhdan then, by definition, $\kappa_\Gamma(S)>0$. Obviously, for a particular essentially nontrivial unitary representation π of Γ, we have by the definition of *inf* that

$$\kappa_\Gamma(S,\pi) \geq \kappa_\Gamma(S).$$

In addition, for a particular unit vector ξ on Γ, by definition of *inf* there exists an $s \in S$ such that

$$\|\pi(s)\xi-\xi\| \geq \kappa_\Gamma(S,\pi).$$

Let π be the regular representation on Γ described above in Chapter 3.3. Then π is:

a) unitary, since $\pi(x)f$ just permutes the entries of any n-tuple f to form a new n-tuple f", where f and f" clearly have the same total when summing entries.

b) essentially nontrivial on Γ in the vector space of n-tuples whose entries sum to zero. This may be justified by the following reasoning:

In the left regular representation on Γ, we have that if $f = (f(h_1), f(h_2), \ldots, f(h_n))$,

then $\pi(h)f = (f(h{\cdot}h_1), f(h{\cdot}h_2), \ldots, f(h{\cdot}h_n))$.

Suppose $f(h_i) \neq f(h_j)$, for some $h_i, h_j \in \Gamma$.
Since Γ is a group, there exists $h \in \Gamma$ such that $h{\cdot}h_i = h_j$.
Then, clearly

$$\pi(h)f \neq f,$$

since the ith entry of f is $f(h_i)$ while the ith entry of $\pi(h)f$ is $f(h{\cdot}h_i) = f(h_j)$ and we know that $f(h_i) \neq f(h_j)$. As such, the only n-tuples which are candidates for not being moved by some $\pi(h)$ are the constant n-tuples. However, the only constant n-tuple in the space of n-tuples whose entries sum to zero is the zero n-tuple, which is excluded from consideration as a vector to be moved in the definition of essentially nontrivial unitary representations. Thus, π is essentially nontrivial.

Partition Γ into (nonempty) sets A and B of group elements and let $a=|A|$ and $b=|B|$. Then we may define the function f from Γ into the integers by

$$f(h) = \begin{cases} b & \text{if } h\epsilon A \\ -a & \text{if } h\epsilon B. \end{cases}$$

Then $g = f(x)/\|f\|$ is a nonconstant unit vector whose entries sum to zero (since the entries of f sum to zero), so it is clear that there exists $s\epsilon S$ such that

$$\|(\pi(s)g)-g\| \geq \kappa_\Gamma(S,\pi) \geq \kappa_\Gamma(S).$$

Noting that

$$\|(\pi(s)g-g\| = \|(\pi(s)f/\|f\|)-(f/\|f\|)\| = \|\pi(s)f-f\|/f\|,$$

we have that

$$\|\pi(s)f-f\|/\|f\| \geq \kappa_\Gamma(S).$$

Since there are $|A|$ entries of value b and $|B|$ entries of value a, straightforward calculation yields

$$
\begin{aligned}
\|f\|^2 &= \sum f(h_i)^2 \\
&= b^2a+(-a)^2b \\
&= b^2a+a^2b \\
&= ab(a+b) = abn,
\end{aligned}
$$

while

$$\|\pi(s)f-f\|^2 = \sum[(\pi(s)f)(h_i-f(h_i)]^2. \qquad (1)$$

Note that if ($h\in A$ and $s*h\in A$) or ($h\in B$ and $s*h\in B$), then $[\pi(s)f](h)=f(h)$. Also, if ($h\in A$ and $s*h\in B$) or ($h\in B$ and $s*h\in A$), then $|\pi(s)f(h)-f(h)| = |b-(-a)| = |(-a)-b| = (a+b) = n$. This means that each vertex $v_i\in V$ associated with its $h_i\in\Gamma$ in the summation of equation (1) will have an edge corresponding to generator s and an edge corresponding to generator s^{-1} (if s is not its own inverse; only one such edge otherwise). These two edges will connect vertex v_i to two other distinct vertices, say v_1 and v_2 in G, by multiplication in Γ. If (v_i,v_1) has one vertex in A and one in B, then it is a "bridge" edge between sets A and B. Similarly, (v_i, v_2) may be a "bridge" edge in the set $E(A,B)$. Each such bridge edge will contribute n^2 to the summation. Then,

$$\sum[(\pi(s)f)(h)-f(h)]^2 = n^2 \cdot E_s(A,B),$$

where $E_s(A,B)$ is the number of s or s^{-1} "bridge" edges in G between sets A and B. Thus, we have

$$\begin{aligned} E_s(A,B) &= \sum[(\pi(s)f)(h)-f(h)]^2/n^2 \\ &= \|\pi(s)f-f\|^2/n^2. \end{aligned}$$

But $\|\pi(s)f-f\|/\|f\| \geq \kappa_r(S)$ and $\|f\|^2 = abn$, so

$$\|\pi(s)f-f\|^2 \geq \kappa_r(S)^2 \cdot abn,$$

and thus

$$\begin{aligned} E_s(A,B) &\geq \kappa_r(S)^2 \cdot abn/n^2 \\ &= \kappa_r(S)^2 \cdot ab/n. \end{aligned}$$

Clearly, the set of vertices in B that are adjacent to set A contains the vertices reached by bridges corresponding to generator s or s^{-1}, since the set of bridges corresponding to s or s^{-1} is a subset of the entire bridge set E(A,B). That is,

$$\partial A_s \subset \partial A,$$

where ∂A_s is defined as the set of vertices in B adjacent to a vertex in A via an edge corresponding to s or s^{-1}, and ∂A is the set of vertices in B adjacent to a vertex in A via any edge in G. But

$$|\partial A_s| \geq \tfrac{1}{2}E_s(A,B),$$

where the ½ allows for the possibility that an adjacent vertex in B may be joined by s and s^{-1} to distinct vertices in A, and therefore contribute two edges to $E_s(A,B)$. Hence,

$$|\partial A| \geq |\partial A_s| \geq \tfrac{1}{2}E_s(A,B) \geq \tfrac{1}{2}[\kappa_r(S)^2 \cdot ab/n]$$

so

$$\begin{aligned} |\partial A|/|A| &\geq \tfrac{1}{2}\kappa_r(S)^2 b/n \\ &= \tfrac{1}{2}\kappa_r(S)^2(n-a)/n \end{aligned}$$

$$= \tfrac{1}{2}\kappa_r(S)^2(1-a/n)$$

$$= \tfrac{1}{2}\kappa_r(S)^2(1-|A|/n)\,.$$

Since the only restriction on A is that A and B partition V, we have that this is true for all proper, nonempty subsets of V.

Then, $c_{max} \geq \tfrac{1}{2}\kappa_r(S)^2$, by definition.

(b) ([Lub], Theorem 3.3.2, pg 65, (iv) ⇒ (i)): Let f be a (necessarily nonconstant) unit vector whose entries sum to zero.

Since Q is an n×n symmetric real matrix, a well-known linear algebra theorem (e.g., [FIS], Theorem 6.20) states that it is orthogonally equivalent to a real diagonal matrix. This implies that we may select an orthonormal basis $\beta=\{x_0,x_1,\ldots,x_{n-1}\}$ of eigenvectors of Q such that, for each i, $Qx_i = \lambda_i x_i$, where $\lambda_{n-1} \geq \lambda_{n-2} \geq \ldots \geq \lambda_1 > \lambda_0$ are the previously found real eigenvalues of Q. Since f is a unit vector whose entries sum to zero, x_0 is not required for producing f as a linear combination of those eigenvectors, as demonstrated by the arguments in Appendix A. Hence, f may be written as $f = a_1x_1 + \ldots + a_{n-1}x_{n-1}$. Then

$$\begin{aligned} Qf &= Q[a_1x_1 + \ldots + a_{n-1}x_{n-1}] \\ &= a_1Qx_1 + \ldots + a_{n-1}Qx_{n-1} \\ &= a_1\lambda_1x_1 + \ldots + a_{n-1}\lambda_{n-1}x_{n-1}. \end{aligned}$$

Because we have an orthonormal basis, $x_ix_j=0$ when $i\neq j$, and $x_i^2=1$ for every i between 1 and n-1. Then, since λ_1 is the smallest nonzero eigenvalue, we have

$$\begin{aligned} Qf \bullet Qf &= a_1^2\lambda_1^2x_1^2 + \ldots + a_{n-1}^2\lambda_{n-1}^2x_{n-1}^2 \\ &\geq a_1^2\lambda_1^2x_1^2 + \ldots + a_{n-1}^2\lambda_1^2x_{n-1}^2 \\ &= \lambda_1^2 \sum a_i^2. \end{aligned}$$

Observing that

$$1 = \|f\| = f \bullet f = \sum a_i^2x_i^2 = \sum a_i^2$$

we have

$$Qf \bullet Qf \geq \lambda_1^2,$$

or,

$$\|Qf\| \geq \lambda_1.$$

Recall that $Q=K-N$, where K is the diagonal matrix with $K_{ii}=k$ and N is the adjacency matrix for the Cayley graph. If π is the regular representation of Γ on the vector space of n-tuples whose entries sum to zero, then the matrix $\pi(s)$ has as entries

$$\pi(s)_{ij} = \begin{cases} 1 & \text{in the ij}^{\text{th}} \text{ entry if } h_i * s = h_j \\ 0 & \text{otherwise.} \end{cases}$$

Therefore, the adjacency matrix N may be viewed as

$$N = \sum_{s\in S} \pi(s),$$

since each entry of 1 is picked up exactly once by the appropriate s. (Each edge of the Cayley graph is affiliated with one element $s\in S$.) Hence

$$\begin{aligned} Qf &= [K-N]f \\ &= [I+I+\ldots+I-\pi(s_1)-\pi(s_2)-\ldots-\pi(s_k)]f \\ &= [I-\pi(s_1)]f +\ldots+ [I-\pi(s_k)]f \\ &= [\pi(e)-\pi(s_1)]f +\ldots+ [\pi(e)-\pi(s_k)]f, \text{ since } \pi(e)=I, \\ &= \pi(e-s_1)f +\ldots+ \pi(e-s_k)f, \text{ since } \pi \text{ is a homomorphism.} \end{aligned}$$

Thus

$$\|Qf\| = \|\pi(e-s_1)f + \ldots + \pi(e-s_k)f\|$$
$$\leq \|\pi(e-s_1)f\| + \ldots + \|\pi(e-s_k)f\|, \text{ by Cauchy-Schwarz.}$$

Since there must exist $s \in S$ such that

$$\|\pi(e-s)f\| = max\{\|\pi(e-s_i)f\|: s_i \in S\}$$

we have

$$\lambda_1 \leq \|Qf\| \leq k\|\pi(e-s)f\| = k\|f-\pi(s)f\|,$$

and, thus

$$\lambda_1/k \leq \|f-\pi(s)f\|.$$

Recall from Theorem 3.3.5 above that every irreducible representation ρ is contained in the regular representation π. Hence, for an f with nonzero entries only in those positions affected by ρ -- that is, in the subspace stable under ρ -- we get that $\rho(s)$ and $\pi(s)$ will produce the identical "moves" on f, and so

$$\|f-\pi(s)f\| = \|f-\rho(s)f\|.$$

But f was an arbitrarily chosen unit vector whose entries sum to zero, and so it is true for particular f's chosen to allow ρ and π to act equivalently on f. Hence, it is always true that

$$\lambda_1/k \leq \|f-\rho(s)f\|$$

when f is chosen in the subspace stable under ρ. That is, when $f \in \hat{H}_\rho$ as defined in the Kazhdan constants.
Hence, we have that $\kappa_r(S,\rho) \geq \lambda_1/k$ for every irreducible ρ, and so

$$\hat{\kappa}_r(S) \geq \lambda_1/k$$

as desired. □

Finally, in a straightforward manner, Theorems 3.2.5 and 3.4.2 may be combined to yield the promised Kazhdan constant bounds on λ_1, $h(G)$, and c_{max}.

THEOREM 3.4.3 ([BdlH], Coro. to Prop. 6, Appdx.): With the notations as above, we have

(a) $\kappa_r(S)^4/(32k) \leq \lambda_1 \leq k\cdot\hat{\kappa}_r(S)$

(b) $\kappa_r(S)^2/4 \leq h \leq k(2\hat{\kappa}_r(S))^{\frac{1}{2}}$

(c) $\frac{1}{2}\kappa_r(S)^2 \leq c_{max} \leq 2k(2\hat{\kappa}_r(S))^{\frac{1}{2}}$.

Proof:

(a) $\lambda_1 \geq h^2/(2k)$, by Theorem 3.2.5, part (d),

$\geq (\frac{1}{4}c_{max}{}^2)/(2k)$, by Theorem 3.2.5, part (b),

$\geq c_{max}{}^2/(8k)$

$\geq [\frac{1}{2}\kappa_r(S)^2]^2/(8k)$, by Theorem 3.4.2, part (a),

$= \kappa_r(S)^4/(32k)$,

thus completing the left-hand inequality; and

$\lambda_1 \leq k\hat{\kappa}_r(S)$, direct from Theorem 3.4.2, part (b).

(b) $\kappa_r(S)^2/4 = \frac{1}{2}(\frac{1}{2}\kappa_r(S)^2)$

$\leq \frac{1}{2}c_{max}$, by Theorem 3.4.2, part (a),

$\leq h$, by Theorem 3.2.5, part (c),

thus completing the left-hand inequality; and

$$k(2\hat{\kappa}_r(S))^{\frac{1}{2}} = (2k^2\hat{\kappa}_r(S))^{\frac{1}{2}}$$

$\geq (2k\lambda_1)^{\frac{1}{2}}$, since $k\cdot\hat{\kappa}_r(S)\geq\lambda_1$, by Theorem 3.4.2,

$\geq h$, since $2k\lambda_1\geq h^2$, by Theorem 3.2.5,

thus completing the right-hand inequality.

(c) $\frac{1}{2}\kappa_r(S)^2 \leq c_{\max}$, from part (a) of Theorem 3.4.2,

thus completing the left-hand inequality; and

$c_{\max} \leq 2k(2\hat{\kappa}_r(S))^{\frac{1}{2}}$, from $h\geq\frac{1}{2}c_{\max}$ and Theorem 3.4.3,

thus completing the right-hand inequality. □

3.5 Evaluating The Kazhdan Bounds

Now that certain bounds have been established by Theorem 3.4.3, it remains to be seen how useful these boundaries are. Clearly, this all depends on how sharp the bounds are and on how well we may evaluate the Kazhdan constants $\kappa_\Gamma(S)$ and $\hat{\kappa}_\Gamma(S)$. Recalling from Definition 3.4.1, we have

(a) $\kappa_\Gamma(S,\pi) = \mathit{inf}\{\mathit{max}\{\|\pi(s)\xi-\xi\|: s\in S\}: \xi\in\mathfrak{H}_\pi \text{ and } \|\xi\|=1\}$

(b) $\kappa_\Gamma(S) = \mathit{inf}\{\kappa_\Gamma(S,\pi): \pi \text{ is essentially nontrivial}\}$

(c) $\hat{\kappa}_\Gamma(S) = \mathit{inf}\{\kappa_\Gamma(S,\pi): \pi \text{ is irreducible and ess. nontr.}\}$.

In general, it is difficult to evaluate these constants, but we can take advantage of the fact that the irreducible representations on any abelian group all have degree (or dimension) 1 over the complex numbers (Theorem 3.3.6, above). This means that such a π represents an abelian group by a group of 1x1 matrices with complex entries; in other words, π sends elements of an abelian group to the complex numbers. Hence, any such π on an abelian group A is a homomorphism of the form

$$\pi: A \mapsto \mathbb{C}^* = \mathbb{C}-\{0\}.$$

(Zero is removed since it has no inverse. $\mathbb{C}^*$ is a group under multiplication of complex numbers.) In addition, since π is unitary, we are restricted to values in $\mathbb{C}^*$ which lie on the unit circle, thereby preserving length when multiplying by a "1-tuple" in the complex numbers.

CASE 1: $A=Z_n$.

Since Z_n is a cyclic group, the representation group in $\mathbb{C}^*$ is fully defined by where π sends a generator of Z_n. In particular, it is fully defined by where π sends $1\epsilon Z_n$. For example, $\pi(\overline{5})$ must be sent to the 5th power of $\pi(\overline{1})$. That is

$$\pi(\overline{5}) = \pi(\overline{1}+\overline{1}+\overline{1}+\overline{1}+\overline{1}) = \pi(\overline{1})\pi(\overline{1})\pi(\overline{1})\pi(\overline{1})\pi(\overline{1}) = [\pi(\overline{1})]^5.$$

Similarly, $\pi(\overline{s})$ must be sent to the s^{th} power of $\pi(\overline{1})$, for every s from 0 through n-1. By the properties of homomorphisms, we know that identity must be sent to identity. Hence

$$1 = \pi(\overline{0}) = \pi(n\cdot\overline{1}) = \pi(\overline{1}+\ldots+\overline{1}) = \pi(\overline{1})\cdots\pi(\overline{1}) = [\pi(\overline{1})]^n$$

thereby implying that $\overline{1}$ must be sent to a number in $\mathbb{C}^*$ whose n^{th} power is 1. Thus there are just n distinct choices for where to send $\overline{1}$, those being the n distinct n^{th} roots of unity. Therefore, there are just n distinct possibilities for irreducible unitary representations of Z_n: they are $\pi_0, \ldots, \pi_{n-1}$, where

$$\pi_k(\overline{1}) \mapsto e^{\iota 2\pi k/n} = \cos(2\pi k/n) + \iota\sin(2\pi k/n).$$

Clearly

$$\begin{aligned}\pi_k(\overline{s}) = [\pi_k(\overline{1})]^s &= [\cos(2\pi k/n) + \iota\sin(2\pi k/n)]^s \\ &= \cos(2\pi k\cdot s/n) + \iota\sin(2\pi k\cdot s/n),\end{aligned}$$

by repeated application of DeMoivre's Theorem. In addition, we may discard π_0 (which sends $\overline{1}$ to 1) since it is the trivial representation, and therefore does not satisfy the condition of essentially non-trivial.

Since we are operating in a 1-dimensional space, the choice for the unit vector ξ is also restricted to a complex number on the unit circle when evaluating $\kappa_r(S,\Pi)$. Fortunately, we can see by the following that we need not evaluate at each of the (infinitely!) many such complex numbers, but instead need only evaluate at $\xi=1$, since all such values yield equivalent results.

Let $\xi=a+b\iota$, arbitrary on the complex unit circle. Then, since $\|c+d\iota\| \equiv |c+d\iota| = (c^2+d^2)^{½}$, we have

$$\begin{aligned}\|\Pi_k(s)\xi-\xi\| = \|(\Pi_k(s)-1)\xi\| &= |(\Pi_k(s)-1)\xi| \\ &= |(\Pi_k(s)-1)a + (\Pi_k(s)-1)b\iota| \\ &= [(\Pi_k(s)-1)^2a^2 + (\Pi_k(s)-1)^2b^2]^{½} \\ &= |(\Pi_k(s)-1)|(a^2+b^2)^{½} \\ &= |(\Pi_k(s)-1)|\cdot 1 \\ &= |\Pi_k(s)-1|.\end{aligned}$$

Thus, the task for evaluating $\hat{\kappa}_r(S)$ for a specific generating set S reduces to finding the value of

$$max_{s\epsilon S} \; |\Pi_k(s)-1|$$

for each of the (finite number of) π_k's, and taking the minimum of these results as the value of $\hat{\kappa}_r(S)$. For a particular $s \in S$, we have

$$\begin{aligned}
|\pi_k(s)-1| &= |\cos(2\pi ks/n) + i\sin(2\pi ks/n)-1| \\
&= [(\cos(2\pi ks/n)-1)^2 + \sin^2(2\pi ks/n)]^{1/2} \\
&= [\cos^2(2\pi ks/n) - 2\cos(2\pi ks/n) + 1 + \sin^2(2\pi ks/n)]^{1/2} \\
&= [2 - 2\cos(2\pi ks/n)]^{1/2} \\
&= 2^{1/2}[1 - \cos(2\pi ks/n)]^{1/2} \\
&= (2^{1/2})^2[\{1 - \cos(2\pi ks/n)\}/2]^{1/2} \\
&= 2|\sin(\pi ks/n)|, \text{ by the half-angle identity.}
\end{aligned}$$

Hence, it is relatively straightforward, though perhaps laborious, to evaluate $\hat{\kappa}_r(S)$ on Z_n for various generating sets S.

EXAMPLE 3.5.1: Z_{21}, with generating set $S=\{\bar{1}\}$.

Evaluating for k=1 through 20 it is clear that this term has its minimum value when k=1 or 20, since these are the two values for which the angle is closest to horizontal on the unit circle. Hence

$$\hat{\kappa}_r(\{\overline{1}\}) = 2\sin(\pi/21) = 2\sin(20\pi/21) \approx .2981.$$

Note that $\hat{\kappa}_r(\{s\}) = 2\sin(\pi/n)$ for **any** single generator generating set, since we always get that

$$ks \equiv 1(\text{mod } n)$$

for **some** value of k between 1 and n-1. ///

Then, for the general case of generating sets on Z_n, we have the following theorem:

THEOREM 3.5.2: For Z_{pq} with generating set $S=\{\overline{p},\overline{q}\}$ where the gcd(p,q)=1, we have that $\hat{\kappa}_r(S) = 2\sin(\pi/a)$, where $a=\mathit{max}\{p,q\}$.

***Proof*:** We have from above that

$$\pi_k(\bar{p}) = \cos(2\pi k/q) + \iota\sin(2\pi k/q)$$

and

$$\pi_k(\bar{q}) = \cos(2\pi k/p) + \iota\sin(2\pi k/p).$$

Also

$$|\pi_k(\bar{p})-1| = 2|\sin(\pi k/q)| \quad \text{and} \quad |\pi_k(\bar{q})-1| = 2|\sin(\pi k/p)|.$$

Observe that when $k=tq$ for some $t\in\mathbb{I}$, $\sin(\pi k/q) = \sin(\pi t) = 0$. Similarly, when $k=tp$, we have $\sin(\pi k/p) = \sin(\pi t) = 0$. In these cases, then, the other generator provides the maximum value of $\kappa_r(S,\pi_k)$. Hence, if we look for:

$$A = \mathit{min}\{2|\sin(\pi k/q)|\text{: k a multiple of p}\}, \text{ and}$$

$$B = \mathit{min}\{2|\sin(\pi k/p)|\text{: k a multiple of q}\},$$

we have that

$$\hat{\kappa}_r(S) \leq \mathit{min}\{A,B\}.$$

To evaluate A: since the closer $\pi k/q$ is to a multiple of π, the smaller $2|\sin(\pi k/q)|$ is, we look for k/q to be as close to an integer as possible while k is a multiple of p. That is, we search for the minimum value of t where

$$k \equiv \pm t \pmod q \text{ and } k \equiv 0 \pmod p.$$

Clearly, there are $q-1$ candidate values of k as a multiple of p to check since k runs from 1 through $(pq-1)$.

Case 1: each value of k that is a multiple of p yields a distinct value modulus q.

Then, since there are only $q-1$ possible distinct values of a number when considered modulus q, each such value is represented. In particular,

$$k \equiv 1 \pmod q$$

for one of the values of k that is a multiple of p. Hence, $|\sin(\pi k/q)| = \sin(\pi/q)$ for one of the values of k.

Case 2: Not all the values of modulus q are represented. Then, for at least two distinct values x and y between 1 and q-1, we have, by the Pigeonhole Principle, that their resulting distinct values of k are equivalent modulus q. That is

$$
\begin{aligned}
& xp-yp \equiv 0 \pmod{q}, \\
\Rightarrow \quad & (x-y)p \equiv 0 \pmod{q}, \\
\Rightarrow \quad & q \text{ divides } (x-y),
\end{aligned}
$$

since $\gcd(q,p)=1$. But $1 \leq |x-y| \leq q-1$, so it is not possible for q to divide (x-y), contradicting the possibility of Case 2.

Hence, Case 1 applies at all times, and we get $\sin(\pi k/q) = \sin(\pi/q)$ for one of the values of k for which k is a multiple of p.

This argument is, of course, equally valid for arriving at the conclusion that $\sin(\pi k/p) = \sin(\pi/p)$ for one of the values of k for which k is a multiple of q. Then

$$\hat{\mathcal{K}}_r(S) \leq \mathit{min}\{2\sin(\pi/q), 2\sin(\pi/p)\},$$

selecting, of course, the value of the term with the larger of q or p. Clearly, though, whenever k is neither a multiple of q or p, we have

$$2|\sin(\pi k/q)| \geq 2\sin(\pi/q) \quad \text{and} \quad 2|\sin(\pi k/p)| \geq 2\sin(\pi/p)$$

since $\pi k/q$ cannot be an angle closer to horizontal than π/q (without actually being horizontal!) thus allowing us to conclude that

$$\begin{aligned} \hat{\mathcal{K}}_r(S) &= \mathit{min}\{2\sin(\pi/q),\ 2\sin(\pi/p)\} \\ &= 2\sin(\pi/a) \end{aligned}$$

where $a=\mathit{max}\{p,q\}$. □

EXAMPLE 3.5.3: Z_{21} with $S=\{\overline{3},\overline{7}\}$.

Then, by Theorem 3.5.2 above, we have

$$\hat{\mathcal{K}}_r(\{\overline{3},\overline{7}\}) = \mathit{min}\{2\sin(\pi/3), 2\sin(\pi/7)\}$$

$$= 2\sin(\pi/7) \approx .8678. \qquad ///$$

Observe that $Z_n = Z_{p_1p_2\cdots p_t}$, all p_i primes, can be generated by any pair of distinct primes $S=\{p_i,p_j\}$, since $\gcd(p_i,p_j)=1$. Then, evaluating the Kazhdan constant for this generating set S is the same as for Z_{pq}, except that the denominator will be the full product of the p_i's. That is

$$\hat{\mathcal{K}}_r(S) = 2\sin(a\pi/[p_1p_2 \ldots p_t]),$$

where $a=\mathit{min}\{p_i,p_j\}$. However, this value of $\hat{\mathcal{K}}_r$ can clearly be raised (improved) by treating our group as Z_{pq} and choosing a generating set $S=\{p,q\}$ where

$$p = p_{i_1}p_{i_2}\cdots p_{i_n} \quad \text{and} \quad q = p_{j_1}p_{j_2}\cdots p_{j_m}$$

such that the p_i's are partitioned into the products forming p and q, and $\mathit{min}\{p,q\}$ is maximized for a partitioning where $\gcd(p,q)=1$. Hence

$$\hat{\mathcal{K}}_r(S) = \mathit{min}\{2\sin(\pi/q), 2\sin(\pi/p)\}$$

which, when $t \geq 4$, will usually produce a higher Kazhdan constant than choosing even the best pair for $S=\{p_i,p_j\}$ since p_i/n and p_j/n will generally both be less than $\mathit{min}\{p/n, q/n\}$.

In a similar fashion, we may deal with groups of the form $Z_{p^n q^m}$, for $\gcd(p,q)=1$, by using the generating set $S=\{p^n, q^m\}$ and evaluating as if our group were Z_{ab}, with $a=p^n$ and $b=q^m$, since we still have that the $\gcd(p^n,q^m) = 1$.

For those cases that don't fit into a nice category, we may still find $\hat{\mathcal{K}}_r(S)$ for Z_n with generating set $S=\{x_1, \ldots, x_t\}$ as follows:

Define $f_k = \mathit{max}\{|kx_i (\mathrm{mod}\ n)|: x_i \in S,\ 0 \leq |kx_i (\mathrm{mod}\ n)| \leq n/2\}$ for each value of k from 1 through n-1. That is, we look for the maximum absolute value of the various kx_i (mod n) where the value modulus n is written in plus form if it is between 0 and n/2 (inclusive) but written in minus form if it would be between n/2 and n-1. For example 9 would be written as

2(mod 7), while 12 would be written as -2(mod 7), not 5(mod 7). This search yields our best case numerator, since

$$\begin{aligned}\kappa_r(S,\pi_k) &= \mathit{max}\{\|\pi_k(x_i)-1\|:\ x_i\in S\}\\ &= \mathit{max}\{2|\sin(\pi x_i k/n)|:\ x_i\in S\}\\ &= 2\sin(f_k\pi/n).\end{aligned}$$

Note that it is unnecessary to consider both x_i and x_j if they are inverses of each other in Z_n. (E.g., for $\overline{3}$ and $\overline{7}$ in Z_{10}.) This is because

$$\begin{aligned}|kx_j(\text{mod } n)| &= |k(n-x_i)(\text{mod } n)|\\ &= |kn-kx_i(\text{mod } n)|\\ &= |-kx_i(\text{mod } n)| = |kx_i(\text{mod } n)|\end{aligned}$$

so they both contribute the same values when evaluating f_k. This implies that $\hat{\kappa}_r(S)=\hat{\kappa}_r(S')$, where S' is the closure of S under inverses. Similarly, it is unnecessary to evaluate using values of k beyond n/2 since

$$\begin{aligned}|kx_i(\text{mod } n)| &= |-kx_i(\text{mod } n)|\\ &= |nx_i-kx_i(\text{mod } n)|\\ &= |(n-k)x_i(\text{mod } n)|,\end{aligned}$$

and hence

$$\begin{aligned}\kappa_r(S,\pi_k) &= 2\sin(f_k\pi/n)\\ &= 2\sin(f_{n-k}\pi/n)\\ &= \kappa_r(S,\pi_{n-k}).\end{aligned}$$

Then we have that

$$\hat{\kappa}_r(S) = \mathit{min}\{2\sin(f_k\pi/n) : k\in\mathbb{I},\ \text{where } 1 \le k \le n/2\}.$$

For small values of n, it is not too tedious to compile this "table" of values to locate the minimum value of f_k. For larger values of n, we resort to computer calculations of this Kazhdan constant, as presented in Appendix B.

EXAMPLE 3.5.4: Z_{21} with various generating sets.

Compiling the table of kx_i(mod 21) we get

TABLE 3.5.5: VALUES OF kx_i(mod 21)

		x_i									
		1	2	3	4	5	6	7	8	9	10
k	1	1	2	3	4	5	6	7	8	9	10
	2	2	4	6	8	10	-9	-7	-5	-3	-1
	3	3	6	9	-9	-6	-3	0	3	6	9
	4	4	8	-9	-5	-1	3	7	-10	-6	-2
	5	5	10	-6	-1	4	9	-7	-2	3	8
	6	6	-9	-3	3	9	-6	0	6	-9	-3
	7	7	-7	0	7	-7	0	7	-7	0	7
	8	8	-5	3	-10	-2	6	-7	1	9	-4
	9	9	-3	6	-6	3	-9	0	9	-3	6
	10	10	-1	9	-2	8	-3	7	-4	6	-5

Then we have

i) for $S=\{\overline{4},\overline{5}\}$ we get the smallest $f_k=4$ when $k=5$, yielding $\hat{K}_r(S) = 2\sin(4\pi/21) \approx 1.127$.

ii) for $S=\{\overline{4},\overline{8}\}$ we get the smallest $f_k=2$ when $k=5$, yielding $\hat{K}_r(S) = 2\sin(2\pi/21) \approx .5895$.

iii) For the best pairs possible ($\{\overline{4},\overline{5}\}$, $\{\overline{2},\overline{8}\}$, $\{\overline{2},\overline{10}\}$, $\{\overline{1},\overline{5}\}$, and $\{\overline{8},\overline{10}\}$) we get the smallest $f_k=4$ in each case, thereby yielding $\hat{K}_r(S) \approx 1.127$.

iv) We can find "triples" of x_i's with smallest $f_k=6$, yielding $\hat{K}_r(S) = 2\sin(6\pi/21) \approx 1.564$.

v) Also, for $S=\{\text{all of } Z_{21}\}$, we get that the smallest $f_k=7$, so $\hat{K}_r(\text{all of } Z_n) = 2\sin(7\pi/21) \approx 1.732$, which agrees with the claim in [BdlH] for Z_n with all of Z_n available for the generating set. ///

CASE 2: A IS ABELIAN, BUT NOT CYCLIC:

Life gets only slightly more complicated when the underlying (finite) group A is abelian, but not cyclic. Since such an abelian group is isomorphic to the direct sum of finite cyclic groups, we have that

$$A \cong Z_{m_1} \oplus Z_{m_2} \oplus \ldots \oplus Z_{m_t}.$$

where $m_1>1$ and $m_1 \mid m_2 \mid \ldots \mid m_t$.

Since A is abelian, all irreducible representations are still 1-dimensional, so that every candidate π used to determine $\hat{\kappa}_r(S)$ must still be a homomorphism of the form

$$\Pi: A \mapsto \mathbb{C}^*.$$

Of course, such a homomorphism is clearly defined by where it sends each member of the standard generating set

$$T = \{(1,0,\ldots,0),(0,1,\ldots,0),\ldots,(0,0,\ldots,1)\}.$$

Hence, the set of homomorphisms sending A to $\mathbb{C}^*$ is the direct product of the sets of homomorphisms sending each of the Z_{m_i} to $\mathbb{C}^*$. I.e., we have

$$\mathrm{Hom}(A,\mathbb{C}^*) = \mathrm{Hom}(Z_{m_1},\mathbb{C}^*)\times \ \ldots\ \times\mathrm{Hom}(Z_{m_t},\mathbb{C}^*).$$

Then the calculation of $\hat{\kappa}_r(S)$ for a specific generating set is again actually fairly straightforward, reducing to the somewhat tedious process of evaluating each possible

$$\kappa_r(S,\Pi_{j_1j_2\cdots j_t}) = max\{|\Pi_{j_1j_2\cdots j_t}(s)-1|: s\in S\}$$

where we must consider each possible variation on $\Pi_{j_1j_2\cdots j_t}$ described by allowing each j_i to run from 1 through m_i. (For example, $\pi_{1,2,2,3}$ would be the representation that sends (1,0,0,0) to (1,0,0,0), (0,1,0,0) to (0,2,0,0), (0,0,1,0) to (0,0,2,0), and (0,0,0,1) to (0,0,0,3). Thus, $\pi_{1,2,2,3}[(2,3,4,5)] = (2,6,8,15)$, where each i^{th} entry would then be reduced modulo Z_{m_i}.) Then

$$\hat{\kappa}_r(S) = min\{\kappa_r(S, \Pi_{j_1j_2\cdots j_t})\} \text{ over all possible } \Pi_{j_1j_2\cdots j_t}.$$

In some specific cases, we may use some knowledge of the situation to "prune" the list of candidate irreducible representations down to a manageable size. To elaborate on this "pruning" process, consider the group $A = Z_n \times Z_m$ where n divides m. Then any irreducible $\pi \in Hom(A, \mathbb{C}^*)$ is of the form

$$\pi_{jk} = \pi_j \times \pi_k$$

where π_j is the standard irreducible representation on the group Z_n into $\mathbb{C}^*$ which sends $\bar{1}$ to $e^{\iota 2\pi j/n}$ and π_k is the irreducible representation on the group Z_m into $\mathbb{C}^*$ which sends $\overline{1}$ to $e^{\iota 2\pi k/n}$. Hence

$$\begin{aligned}\pi_{jk}[(1,1)] &= \pi_{jk}[(1,0)+(0,1)]\\ &= \pi_{jk}(1,0)\pi_{jk}(0,1)\\ &= \pi_j(\overline{1})\pi_k(\overline{1})\\ &= e^{\iota 2\pi j/n}\cdot e^{\iota 2\pi k/m}\\ &= e^{\iota 2\pi[j/n + k/m]}.\end{aligned}$$

Then, by previous calculations,

$$|\pi_{jk}(1,1)-1| = 2|\sin(\pi[j/n + k/m])|$$

and, for general $s=(a,b)$ in the generating set, we have

$$|\pi_{jk}(a,b)-1| = 2|\sin(\pi[ja/n + kb/m])|.$$

Let $c\in N$ be such that $cn=m$.

Then

$$ja/n + kb/m = (jac+kb)/m.$$

Since the numerator is a positive integer, we have that the smallest nonzero value possible for this quotient is 1/m. that is

$$\begin{aligned}|\pi_{jk}(a,b)-1| &= 2|\sin(\pi[ja/n + kb/m])| \\ &\geq 2\sin(\pi/m) \text{ when } m \nmid (jac+kb).\end{aligned}$$

If, for a particular π_{jk}, we have that $\pi_{jk}(s)=1$ for every $s\in S$, then π_{jk} is the trivial representation and is not to be considered when calculating $\hat{\kappa}_r(S)$. Hence, for every π_{jk} we need to consider, for at least one $s\in S$ we have that $\pi_{jk}(s)\neq 1$. Then

$$|\pi_{jk}(s)-1| \geq 2\sin(\pi/m).$$

Clearly, in the course of considering all the possible π_{jk}'s, we must consider those when $j=0$. Then

$$\begin{aligned}|\pi_{0k}(a,b)-1| &= 2|\sin(\pi[0a/n + kb/m])| \\ &= 2|\sin(kb/m)|.\end{aligned}$$

If $S=\{(1,0),(0,1)\}$, the standard generating set, then $b=1$ and we are done since for $k=1$ we have found a representation π_{01} for which the value of

$$\begin{aligned}\kappa_r(S,\pi_{01}) &= \mathit{max}\{|\pi_{01}(1,0)-1|,|\pi_{01}(0,1)-1|\} \\ &= |\pi_{01}(0,1)-1| \\ &= 2\sin(\pi/m)\end{aligned}$$

is the minimum possible $\kappa_r(S,\pi_{jk})$ under consideration. Then

$$\hat{\kappa}_r(S) = 2\sin(\pi/m).$$

By extension, if the generating set S contains only generators of the form $s=(a,b)$ for one fixed b where $\gcd(b,m)=1$, while all other generators are of the form $(a,0)$, then we can be sure that, for some value of k between 1 and $m-1$, we will have by the Pigeonhole Principle that $kb \equiv 1(\mathrm{mod}\ m)$, meaning that we would have

$$\hat{\kappa}_r(S) = 2\sin(\pi/m).$$

In the general abelian case of $A \cong Z_{m_1} \oplus Z_{m_2} \oplus \ldots \oplus Z_{m_t}$ with the standard generating set or one with the same such restrictions on the t^{th} entries of the generators, we get by the same reasoning that

$$\hat{\kappa}_\Gamma(S) = 2\sin(\pi/m_t).$$

Hence, we have established methods of evaluating $\hat{\kappa}_\Gamma(S)$ for various abelian groups, particularly the cyclic group Z_n. The difficulty encountered here is that we have not yet developed a simple way to calculate $\kappa_\Gamma(S)$, so we do not have the lower bounds of Theorem 3.4.3 in place. However, there is a relation between $\hat{\kappa}_\Gamma(S)$ and $\kappa_\Gamma(S)$ that provides a lower bound on $\kappa_\Gamma(S)$ which will give us a starting point. The relationship is provided here as the following theorem:

THEOREM 3.5.6 ([BdlH], Proposition 2): For any finite generating set S of the finite group Γ, we have that

$$\hat{\kappa}_\Gamma(S) \geq \kappa_\Gamma(S) \geq \hat{\kappa}_\Gamma(S)/[|S|^{½}].$$

***Proof*:** The left-hand inequality is obvious from the definitions of the two Kazhdan constants, since the set of irreducible representations to be considered for $\hat{\kappa}_\Gamma(S)$ is a subset of the representations to be considered for $\kappa_\Gamma(S)$. For the right-hand inequality, we prove as follows:

Suppose there exists π, a unitary representation of Γ such that

$$\kappa_\Gamma(S,\pi) < \hat{\kappa}_\Gamma(S)/[|S|^{½}].$$

We shall show that this forces π to be essentially nontrivial.

By the supposition there exists a unit vector ξ such that

$$\|\xi\|^2 = 1$$

and

$$(max_{s\in S}\|\pi(s)\xi-\xi\|)^2 < (\hat{\kappa}_\Gamma(S))^2/|S|.$$

By complete reducibility of π, we may break it down into its irreducible subrepresentations. That is, we find the direct sum of nontrivial subspaces of the vector space V over which π is invariant. We denote this family of subspaces by $(V_\omega)_{\omega\in\Omega}$. Using an appropriate basis for this direct sum, we may write ξ as the sum of vectors in each subspace. That is

$$\xi = (\xi_\omega)_{\omega\in\Omega},$$

where $\xi_\omega \in V_\omega$ for every $\omega\in\Omega$.

Then, for $\xi=(f_1,\ldots,f_n)$ we have

$$\begin{aligned}1 &= \|\xi\|^2\\ &= (f_1,\ldots,f_n)(f_1,\ldots,f_n)\\ &= \sum f_i^2\\ &= \sum(\text{clusters of } f_i^2 \text{ arranged as desired})\\ &= \sum\|\xi_\omega\|^2.\end{aligned}$$

In addition, we may see that, for a particular $s\in S$

$$\begin{aligned}\|\pi(s)\xi-\xi\|^2 &= \|(\pi(s)f_1-f_1,\ldots,\pi(s)f_n-f_n)\|^2\\ &= [(\pi(s)f_1-f_1]^2 +\ldots+ [(\pi(s)f_n-f_n]^2\end{aligned}$$

$$= \sum_{\omega\in\Omega} \|\pi_\omega(s)\xi_\omega - \xi_\omega\|^2,$$

since again the summation of terms may be grouped as desired. Hence, for all $s\in S$, we have that

$$\sum_{\omega\in\Omega} \|\pi_\omega(s)\xi_\omega - \xi_\omega\|^2 = \|\pi(s)\xi - \xi\|)^2 < (\hat{K}_r(S))^2/|S|.$$

Define for each $s\in S$

$$\Omega_s = \{\omega\in\Omega\colon \|\pi_\omega(s)\xi_\omega - \xi_\omega\| \geq \hat{K}_r(S)\|\xi_\omega\|\}.$$

Then

$$\begin{aligned}(\hat{K}_r(S))^2/|S| &> \sum_{\omega\in\Omega} \|\pi_\omega(s)\xi_\omega - \xi_\omega\|^2 \\ &\geq \sum_{\omega\in\Omega_s} \|\pi_\omega(s)\xi_\omega - \xi_\omega\|^2 \\ &\geq (\hat{K}_r(S))^2 \sum_{\omega\in\Omega_s} \|\xi_\omega\|^2\end{aligned}$$

and so

$$1/|S| > \sum_{\omega\in\Omega_s} \|\xi_\omega\|^2.$$

Set $\Omega' = \bigcup_{s\in S} \Omega_s$. Since the sum of the sizes of all subsets Ω_s must be greater than or equal to the size of the union of all those subsets, we have $|\Omega'| \leq \sum_{s\in S} |\Omega_s|$. Then

$$\begin{aligned}\sum_{\omega\in\Omega'} \|\xi_\omega\|^2 &\le \sum_{s\in S}\sum_{\omega\in\Omega_s} \|\xi_\omega\|^2 \\ &< |S|\cdot 1/|S| \\ &= 1 \\ &= \sum_{\omega\in\Omega} \|\xi_\omega\|^2 .\end{aligned}$$

Since this last is a strict inequality, it must be the case that Ω' is a proper subset of Ω. Then the set $\Omega\text{-}\Omega'$ is non-empty. So, by the definition of the sets Ω_s, there exists a $\omega\in\Omega$ for which, no matter what generator $s\in S$ is chosen, it is true that

$$\|\pi_\omega(s)\xi_\omega - \xi_\omega\| < \hat{\kappa}_r(S)\|\xi_\omega\| .$$

That is

$$max_{s\in S}\ \|\pi_\omega(s)\xi_\omega - \xi_\omega\| < \hat{\kappa}_r(S)\|\xi_\omega\| .$$

Then

$$\begin{aligned}\hat{\kappa}_r(S) &> (1/\|\xi_\omega\|)\,max_{s\in S}\ \|\pi_\omega(s)\xi_\omega - \xi_\omega\| \\ &= max_{s\in S}\ \|\pi_\omega(s)\xi_\omega' - \xi_\omega'\|\end{aligned}$$

where ξ_ω' is the unit vector equal to $\xi_\omega/\|\xi_\omega\|$.

Since ξ_ω' is a unit vector, by the definition of $\hat{\kappa}_\Gamma(S)$ we may conclude that π_ω cannot be an essentially nontrivial representation. That is, π itself is not essentially nontrivial. Hence, no π satisfying the supposition may be essentially nontrivial, implying that all π's that are essentially nontrivial satisfy the statement

$$\kappa_\Gamma(S,\pi) \geq \hat{\kappa}_\Gamma(S)/|S|^{1/2}.$$

But $\kappa_\Gamma(S) = \mathit{inf}\{\kappa_\Gamma(S,\pi) : \pi \text{ is essentially nontrivial}\}$, so we have that

$$\kappa_\Gamma(S) \geq \hat{\kappa}_\Gamma(S)/|S|^{1/2},$$

thereby proving the right-hand inequality. □

In order to make best use of this theorem for evaluating the lower bound for the case of our Cayley graphs on abelian groups Γ, observe that, for every irreducible nontrivial π we must evaluate, we have

$$\kappa_r(S,\pi) \le \kappa_r(S',\pi)$$

where S' is the closure under inverses of S. This is clearly the case since $S \subseteq S'$ and

$$max\{|\pi(s)-1|: s \in S\} \le max\{|\pi(s)-1|: s \in S'\}.$$

Then we have $\hat{\kappa}_r(S) \le \hat{\kappa}_r(S')$ and $\kappa_r(S) \le \kappa_r(S')$ since $\hat{\kappa}_r(S)$ and $\hat{\kappa}_r(S')$ evaluate for the same set of irreducible π's, while $\kappa_r(S)$ and $\kappa_r(S')$ also evaluate over their same set of π's.

Combining theorem 3.4.6 and these inequalities, we get that

$$\hat{\kappa}_r(S') \ge \hat{\kappa}_r(S) \ge \kappa_r(S) \ge \hat{\kappa}_r(S)/|S|^{½} \quad (\ge \hat{\kappa}_r(S)/|S'|^{½})$$

and so

$$\kappa_r(S') \ge \kappa_r(S) \ge \hat{\kappa}_r(S)/|S|^{½} \quad (\ge \hat{\kappa}_r(S)/|S'|^{½}).$$

Thus, when considering a Cayley graph with a generating set S' (since a Cayley graph requires that the generating set be closed under inverses) we can conclude that

$$\kappa_r(S') \geq \hat{\kappa}_r(S)/|S|^{1/2},$$

where S is a "stripped-down" version of S' with the extraneous inverses cast out. This allows us to use the usually smaller divisor of $|S|^{1/2}$, rather than $|S'|^{1/2}$, thereby improving the lower limits of Theorem 3.4.3. In addition (as observed above for the more limited case of Z_n) no ground is lost in the general abelian case by evaluating the bounds using $\hat{\kappa}_r(S)$ rather than $\hat{\kappa}_r(S')$, since we have

$$|\sin(x)| = |\sin(-x)|$$

$$\Rightarrow \quad 2|\sin(\pi[j_1s_1 + \ldots + j_ts_t]/(j_1j_2\cdots j_t))|$$

$$= 2|\sin(\pi[j_1(-s_1) + \ldots + j_t(-s_t)]/(j_1j_2\cdots j_t))|$$

$$\text{for every } s=(s_1,\ldots,s_t)\in S$$

$$\Rightarrow \quad \|\pi(s)-1\| = \|\pi(-s)-1\|$$

$$\Rightarrow \quad \|\pi(s)-1\| = \|\pi(s^{-1})-1\|$$

$$\Rightarrow \quad \mathit{max}\{\|\pi(s)-1\|:\ s\in S\} = \mathit{max}\{\|\pi(s)-1\|:\ s\in S'\}$$

$$\Rightarrow \quad \kappa_r(S,\pi) = \kappa_r(S',\pi) \text{ for every candidate } \pi$$

$$\Rightarrow \quad \hat{\kappa}_r(S) = \hat{\kappa}_r(S').$$

3.6 How Useful Are The Bounds of Theorem 3.4.3?

Recall from Theorem 3.4.3 that, when G is a Cayley graph on the group Γ with generating set $S=S^{-1}$ and $|S|=k$, we have that

(a) $\kappa_\Gamma(S)^4/(32k) \leq \lambda_1(G) \leq k\hat{\kappa}_\Gamma(S)$

(b) $\kappa_\Gamma(S)^2/4 \leq h(G) \leq k[2\hat{\kappa}_\Gamma(S)]^{1/2}$

(c) $\frac{1}{2}\kappa_\Gamma(S)^2 \leq c_{max}(G) \leq 2k[2\hat{\kappa}_\Gamma(S)]^{1/2}$.

Then it would seem appropriate to select a graph with small enough n that we can examine its structure to find the constants λ_1, h, and c_{max}, and compare their actual values to the bounds supplied by Theorem 3.4.3. Consider the Cayley graph G of Z_{21} with generating set $S=\{\overline{3},\overline{7}\}$. (Note that S' = $\{3,7,3^{-1},7^{-1}\}$ = $\{3,7,18,14\}$, so k=4.) By physical construction and examination of its structure, we have $h(G)=2/3$ and $c_{max}(G)=1.05$ (case when A=20 vertices $\Rightarrow c_{max}$ = $n/(n-1)$ = 21/20 = 1.05). By the Lovasz algorithm, we get $\lambda_1(G)$ = 0.753.

Earlier in Example 3.5.3 we calculated that

$\hat{\kappa}_r(S)\approx 0.8678$, so $\kappa_r(S) \geq \hat{\kappa}_r(S)/2^{½} \approx 0.6136$ and Theorem 3.4.3 yields:

(a) $$(.6136)^4/[32\cdot 4] \leq \lambda_1 \leq 4(.8678)$$

or $$1.107\times 10^{-3} \leq \lambda_1 \leq 3.4712.$$

(b) $$(.6136)^2/4 \leq h \leq 4(2\cdot 0.8678)^{½}$$

or $$0.0941 \leq h \leq 5.270.$$

(c) $$½(.6136)^2 \leq c_{max} \leq 8(2\cdot 0.8678)^{½}$$

or $$0.1882 \leq c_{max} \leq 10.54.$$

Certainly the bounds seem rather soft. Although we don't need part (a) to evaluate λ_1 for abelian groups (since we have the Lovasz algorithm) the spread on the ratio of upper bound to lower bound is of order 3000. This is indicative of how poor these bounds are. Parts (b) and (c), though better, are still not very good, providing a spread on the same ratio of order 56. Examining the spread produced for Z_{60} and Z_{199} produced the following results, where the generating sets used were those which provided the best (highest) value of $\hat{\kappa}_r(S)$ for generating sets of that size:

Z_{60} $S=\{1,7\}$, $\hat{\kappa}_r(S) = .717$ spread for (a): 5500
spread for (b) & (c): 74

Z_{60} $S=\{1,4,13\}$, $\hat{\kappa}_r(S)=1.26$ spread for (a): 3400
spread for (b) & (c): 54

Z_{60} $S=\{1,2,6,18\}$, $\hat{\kappa}_r(S)=1.62$ spread for (a): 3850
spread for (b) & (c): 88

Z_{199} $S=\{1,14\}$, $\hat{\kappa}_r(S)=.4384$ spread for (a): 24000
spread for (b) & (c): 156

Z_{199} $S=\{1,6,31\}$, $\hat{\kappa}_r(S)=.9404$ spread for (a): 12300
spread for (b) & (c): 111

The authors [BdlH] themselves report that part (a) provides an estimate of that is "rather bad" when comparing the bounds on λ_1 for the symmetric group on n letters with a generating set containing all transpositions. For appropriate constants c_1, c_2, they show

$$c_1 n^{-6} \leq \lambda_1 \leq c_2 n^{3/2}$$

while the exact value is known to be $\lambda_1 = n$. We may be heartened by the fact that the actual values for the constants fall within the boundaries provided, but that is about all the help we can get from these bounds.

Also of interest is to consider what happens to the lower bounds as n gets very large. That is, what is the effect on the Kazhdan constant as n increases? If we fix the size of the generating set, does the Kazhdan constant approach zero as n gets very large, or does it have some lower bound? If it were to have a lower bound, this would be useful information, since we would then know that the expanding constant c_{max} has a lower bound for the class of Z_n groups, indicating that they have some favorable traits for keeping diameter small.

Consider Z_n and the generating set fixed at $|S'|=4$ (includes inverses). Using the set $S=\{1,x\}$, where $x=\lfloor n^{1/2}\rfloor$, the truncated square root of n, we get $f_k=x$, and thus

$$\hat{\kappa}_r(S) = 2\sin(x\pi/n).$$

This is easily seen by considering the following cases for ranges of k:

$k=1$:	$\lvert 1\cdot x(\pm\text{mod } n)\rvert = x$, so $f_k=x$.
$2 \le k \le (x-1)$:	$\lvert kx(\pm\text{mod } n)\rvert \ge x$, so $f_k \ge x$.
$x \le k \le n/2$	$\lvert k\cdot 1(\pm\text{mod } n)\rvert = k \ge x$, so $f_k \ge x$.

Clearly, we may conclude that the generating set $S=\{1,y\}$ yields $f_k=y$ for all y such that $2 \le y < x$, and hence produces an inferior value of $\hat{\kappa}_r(S)$ to that of $S=\{1,x\}$. Unfortunately, moderate efforts using number theory to continue this argument have not succeeded, so we have no conclusive proof that the generating set $S=\{1,x\}$ is the best we can do for $|S'|=4$. However, computer calculations for values of n from 1 through 200, as well as for a number of scattered higher values of n, have without fail produced this generating set as the optimum for producing the highest value of f_k and hence $\hat{K}_r(S)$. Should this indeed be the optimum value for $\hat{K}_r(S)$, then it is clear that there would be no lower bound on $\hat{K}_r(S)$ for $|S'|=4$ since we would have that

$$\hat{K}_r(S) = 2\sin(\pi/n^{1/2})$$

which tends toward zero as n gets very large.

Studying the shape of some graphs of $\hat{K}_r(S)$ vs. $|S|$ would appear to be a useful way to get some feel for this subject as to the behavior of $\hat{K}_r(S)$ as n increases. Figures 3.6.1 and 3.6.2 and their accompanying tables (below) will help in this regard. Figure 3.6.1 led initially to an exploration of whether the first jump from $|S|=1$ to $|S|=2$ always makes it to the halfway point for $|S|=n$, and then to an exploration of the weaker supposition as to whether it is always the largest jump. However, the slope of the graph of Z_{1000} (Figure 3.6.2) makes clear that neither of these are the case. (It is also not true for the graph of Z_{70}, though the values of the first jump and the second jump are quite close.) Also clear is the fact that these curves are not "s-shaped", so there does not appear to be an optimum choice for $|S|$ to get the most Kazhdan constant for your "buck" (size of $|S|$). Finally, observing that for fixed value of $|S|$ and increasing n we have a dwindling value for $\hat{\kappa}_r(S)$, we visually confirm that there will not likely be a lower bound on $\hat{\kappa}_r(S)$.

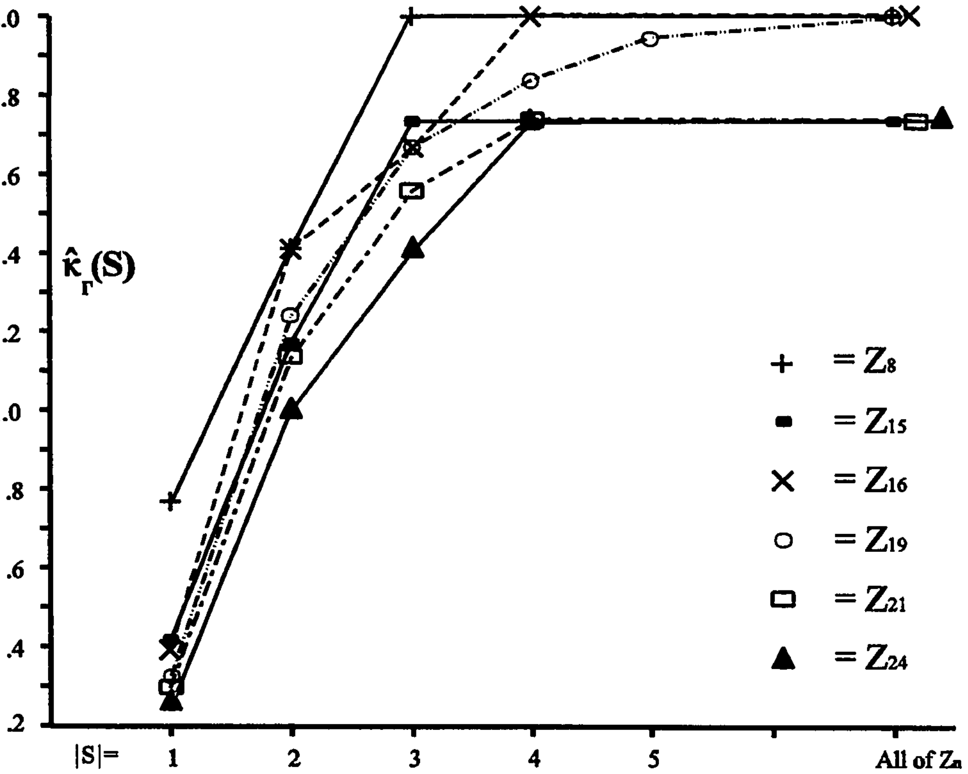

Figure 3.6.1: $\hat{\kappa}_r(S)$ for Various Z_n as a Function of Size of Generating Set S. Vertical axis is $\hat{\kappa}_r(S)$, horizontal axis is |S|. Note that the generating sets indicated are the smallest in size of the possible generating sets resulting in an improvement over the previous Kazhdan constant produced by a smaller generating set. Note also that these generating sets do not contain inverses.

TABLE FOR FIGURE 3.6.1: VALUES OF $\hat{\kappa}_T(S)$ AND f_k FOR VARIOUS Z_n AS SHOWN IN FIGURE 3.6.1

	Z_8		Z_{15}		Z_{16}		Z_{19}		Z_{21}		Z_{24}	
|S|	f_k	$\hat{\kappa}_T$	f_k	$\hat{\kappa}_T$	f_k	$\hat{\kappa}_T$	f_k	$\hat{\kappa}_T$	f_k	$\hat{\kappa}_T$	f_k	$\hat{\kappa}_T$
1	1	.765	1	.416	1	.393	1	.329	1	.298	1	.261
2	2	1.41	3	1.18	4	1.41	4	1.23	4	1.13	4	1.00
3	4	2.00	5	1.73	5	1.66	6	1.67	6	1.56	6	1.41
4					8	2.00	7	1.83	7	1.73	8	1.73
5							8	1.94				
all	4	2.00	5	1.73	8	2.00	9	1.99	7	1.73	8	1.73

Note that the term "all" implies that all $s \in Z_n$ are represented in S. Blank areas indicate that no further gains in f_k may be obtained by a larger set of generators.

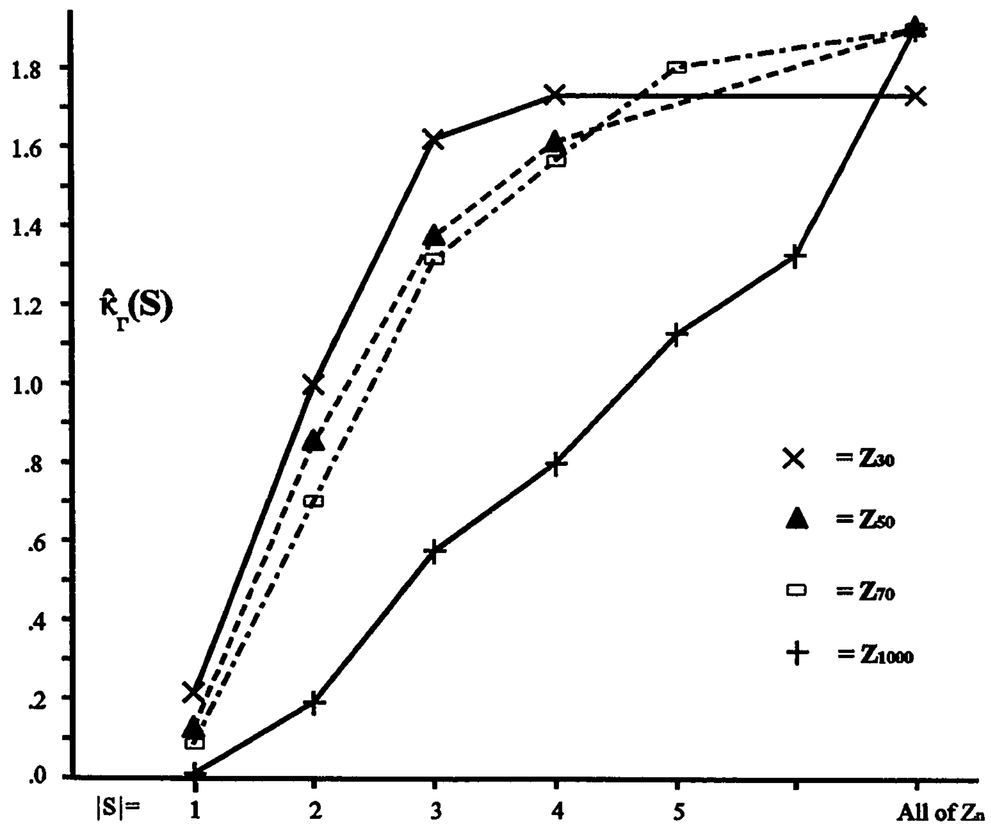

Figure 3.6.2: $\hat{\kappa}_r(S)$ for Various Z_n as a Function of Size of Generating Set S. Vertical axis is $\hat{\kappa}_r(S)$, horizontal axis is |S|. Note that, with the exception of Z_{1000}, the generating sets indicated are the smallest in size of the possible generating sets resulting in an improvement over the previous Kazhdan constant produced by a smaller generating set. Z_{1000} is done by an approximation program which uses the previous best generating set to find the best set with one more generator added on. Note also that these generating sets do not contain inverses.

TABLE FOR FIGURE 3.6.2: VALUES OF $\hat{\kappa}_\Gamma(S)$ AND f_k FOR VARIOUS Z_n, AS SHOWN IN FIGURE 3.6.2

	Z_{30}		Z_{50}		Z_{70}		Z_{1000}	
$\|S\|$	f_k	$\hat{\kappa}_\Gamma$	f_k	$\hat{\kappa}_\Gamma$	f_k	$\hat{\kappa}_\Gamma$	f_k	$\hat{\kappa}_\Gamma$
1	1	.209	1	.126	1	.090	1	.006
2	5	1.00	7	.852	8	.703	31	.194
3	9	1.62	12	1.37	16	1.32	93	.576
4	10	1.73	15	1.62	20	1.56	130	.794
5					25	1.80	190	1.12
6							230	1.32
all*	10	1.73	20	1.90	28	1.90	400	1.90

*Note that "all" implies all $s \in Z_n$ are represented in S. Blank areas indicate either that no further gains in f_k may be obtained for a larger generating set, or that no larger generating sets were tested due to excessive calculation time.

CHAPTER 4

EIGENVALUES AND THE EXPANSION CHARACTERISTICS OF A GRAPH

4.1 Introduction

We found in the previous chapter that the Kazhdan constants were able to supply us with bounds on the expansion characteristics of a graph as described by the constant c_{max}. Though we found that such bounds were not particularly useful for the set of graphs constructed from groups of form Z_n, the study of other bounds on expansion-related constants of Cayley graphs seemed to offer some promise.

There is extensive literature on the subject of families of graphs with good expansion characteristics, and on methods of construction and evaluation of such graphs.

This chapter again explores the relationship of λ_1 with an expansion-related constant of the underlying graph.

4.2 Some General Knowledge About the Spectrum of A(G)

First, it is necessary to develop several facts (used here and in Chapter 2) about the eigenvalues (spectrum) of the standard adjacency matrix A(G) of the Cayley graph G. Recall that A(G) was defined as the nxn symmetric matrix whose entries are given by

$$A(G)_{ij} = A(G)_{ji} = \begin{cases} +1 & \text{if } (v_i, v_j) = (v_j, v_i) \in E(G), \\ & \text{the edge set of } G \\ 0 & \text{otherwise.} \end{cases}$$

Note that this definition sets each entry of the diagonal to zero, since a vertex is not adjacent to itself. Then we have the following theorem:

THEOREM 4.2.1 ([Sch], Proposition 2.1): Let $G=(V,E)$ be a finite, simple k-regular graph, with $|V|=n$, and let $m(\xi)$ denote the multiplicity of the eigenvalue $\xi \in \text{spec}(A(G))$. Then

(a) $|\xi| \leq k$, for every $\xi \in \text{spec}(A(G))$.

(b) $k \in \text{spec}(A(G))$.

(c) G is connected if and only if $m(k)=1$.

(d) G is bipartite if and only if $-k \in \text{spec}(A(G))$, in which case $m(\xi)=m(-\xi)$ for all $\xi \in \text{spec}(A(G))$.

Proof:

(a): Observe that each row of A(G) has d ones in it, since G is k-regular. Suppose $A(G)f = \xi f$; i.e., that ξ is the eigenvalue for some eigenvector $f = (f_1, f_2, \ldots, f_n)$. Then there exists f_i, the i^{th} entry of f, such that

$$|f_i| = \mathit{max}\{|f_j|: 1 \leq j \leq n\}.$$

Since there are k ones in each row of A(G), there are k ones in the i^{th} row of A(G), and so the absolute value of the i^{th}

entry of A(G)f is

$= |f_{s_1} + f_{s_2} + \ldots + f_{s_k}|$, these values being the k entries of f picked up by the ones in the i_{th} row,

$\leq |f_i + f_i + \ldots + f_i|$, since $|f_i| = \mathit{max}\{|f_j|: 1 \leq j \leq n\}$,

$= k|f_i|$.

But, the i_{th} entry of A(G)f is also ξf_i, so

$$|\xi|\cdot|f_i| \leq |\xi f_i| \leq k|f_i|,$$

implying that

$$|\xi| \leq k.$$

(b): Observe that if f = (c,c,c,...,c) is a constant vector, then

$$A(G)f = kf,$$

since the i^{th} entry of df is the sum of k c's, for every value of i. Then f is an eigenvector of A(G), and k is its associated eigenvalue. Hence, by definition, $k \in \text{spec}(A(G))$.

(c):

($\Leftarrow$) Suppose G is **not** connected.

Then G consists of a set of subgraphs, each $G_i=(V_i,E_i)$ of which is a connected k-regular graph with $|V_i|=a_i$, with V partitioned by the V_i's and E partitioned by the E_i's. Then the adjacency matrix A(G) consists of block submatrices A_i along the diagonal (with zeros everywhere else) where each A_i is $a_i \cdot a_i$ and is identical in form to the adjacency matrix of the corresponding subgraph G_i.

Consider the function $g = (c,\ldots,c,0,\ldots,0)$, where there are a_1 consecutive c's. Then we have

$$A(G)g = kg,$$

since the first a_1 rows of A(G) each have all k of their ones packed into the first a_1 columns, so k c's are summed to yield each of the first a_1 entries of A(G)g; since all of the entries of rows a_1+1 through n are zero in the first a_1 columns, each of these rows provide only zero in A(G)g.

Hence, we have found an eigenvector g (with corresponding eigenvalue of k) which is not a constant function, and so the dimension of k's eigenspace is at least two. That is, $m(k)\geq 2$, and so, clearly, $m(k)\neq 1$.

($\Rightarrow$) Suppose $m(k)\geq 2$.

Then there exists a nonconstant function g such that $A(G)g=kg$. Then, as in part (a), there exists nonzero g_i, an entry of g with corresponding vertex v_i, such that

$$|g_i| = \mathit{max}\{|g_j|: 1 \leq j \leq n\}.$$

But every vertex in G is adjacent to only k vertices, so for the sum of v_i's k vertices to yield kg_i, each of those vertices must have a corresponding entry in g equal to that of g_i. In order for that to be possible, each of **those** vertices may only be adjacent to vertices whose corresponding entries in g are equal to g_i. By continuing this reasoning, it is clear that all vertices connected (with a path) to v_i must have entries in g equal to g_i. Then, in order for g to be nonconstant, there must be another "cluster" of vertices **not** connected to v_i, each vertex of which must have the same value of its corresponding entry in g as the other vertices in its "cluster", and that value must be different from g_i.

Hence, G has at least two unconnected clusters; that is, G is not connected, since it has subsets of vertices which are only adjacent to vertices within their own subset.

(d):

($\Rightarrow$) Suppose G is bipartite by partitioning into vertex subsets A and B.

Clearly, a count of edges leaving set A must be the same as the count of edges entering set B. But G is bipartite on sets A and B, so there are no edges internal to set A nor internal to set B. Since G is also k-regular, each vertex in A and each vertex in B must have k adjacent edges, all of which bridge the gap between the two subsets. That is, a vertex in A is adjacent only to vertices in B, and vice versa.

Define a function g on the n vertices by

$$g(v) = \begin{cases} +c & \text{if } v \in A \\ -c & \text{if } v \in B. \end{cases}$$

Then

$$A(G)g = -kg,$$

since the i^{th} row of A(G) will contribute a sum of k "-c's" if $g_i=+c$, or will contribute k "+c's" if $g_i=-c$, due to the adjacency structure of sets A and B; hence, the i_{th} entry of A(G)g is $-kg_i$. Thus, -k is an eigenvalue, by definition.

($\Leftarrow$) Suppose $-k \in \text{spec}(A(G))$.

Without loss of generality, we may assume that G is connected, since the argument may otherwise be given for each subgraph G_i.

By the same argument used in the proof of part (c), any candidate eigenvector g with -k as its eigenvalue must consist of entries whose absolute values are the same for all vertices in a "cluster". Since G is assumed to be connected, this means that every vertex in G has the same absolute value of corresponding entry in g. That is, $|g_i|=|g_j|$ for every i,j from 1 through n. In addition, for the i^{th} entry of A(G)g to be $-kg_i$, each of the k vertices adjacent to v_i must have been assigned the value $-g_i$. In turn, each of those vertices must have their adjacent vertices assigned $+g_i$, and so on.

Since there are only two choices of signs, we must be able to partition the vertices into two sets based on adjacency. Thus, by definition, G is bipartite.

Clearly, for each separate bipartite subgraph, there exists a different nonconstant function (n-tuple) with eigenvalue -k, so we get that m(k)=m(-k).

Suppose that G is bipartite on the partition of vertices into sets A and B.

Let g be an eigenvector with eigenvalue ξ. Then define the function g' by

$$g'(v) = \begin{cases} g(v) & \text{if } v \in A \\ -g(v) & \text{if } v \in B. \end{cases}$$

Then the i^{th} entry of A(G)g' will be:

$-\xi g(v_i)$ if $v_i \in A$, since summing the opposite of all the same terms as for A(G)g;

$\xi g(v_i)$ if $v_i \in B$, since summing all the same terms as for A(G)g.

Hence

$$A(G)g' = -\xi g',$$

and thus, by definition, $-\xi$ is an eigenvalue of g'. Therefore, whenever ξ is an eigenvalue for a function g on the vertices of a bipartite G, we have shown that $-\xi$ is an eigenvalue for the function g'. That is

$$m(\xi) = m(-\xi), \text{ for every } \xi \in \mathrm{spec}(A(G)). \qquad \square$$

4.3 Various Expansion-Related Characteristics of a Graph

The diameter of a graph may be loosely interpreted as a measure of how close every vertex in that graph "lives" to all the other vertices. Hence, it makes sense to explore characteristics of graphs which are related to the density of the "neighborhood" surrounding a vertex or set of vertices. If it may be shown that any such set of vertices has a relatively large set of new neighbors, then the path length from a vertex to any other vertex should (on average) be relatively short.

Two such characteristics have already been introduced into the discussion in Chapter 3: the expanding constant c_{max} of a graph G, which is related to how many "new neighbor" vertices are in the boundary region (all new adjacent vertices) of any given subset of G's vertices, and the Cheeger constant h(G), which is essentially its edge-related equivalent. At this point, we introduce several more expansion-related concepts, and explore the nature of their roles in helping to understand the structures of certain graphs.

DEFINITION 4.3.1: Let $G=G(V,E)$ be a graph on $|V|=n$ vertices and let A be a nonempty subset of V. Then the **neighborhood N(A) of A** is defined by

$N(A) = \{v \in V$: v is adjacent in G to some vertex in A$\}$. □

Note that N(A) includes all vertices in ∂A (the boundary of A, Definition 3.2.1), as well as any vertices in A that are adjacent to another vertex in A. Hence, the neighborhood is at least as large as the boundary, but may not include every vertex of A since it is possible that the only neighbors of a vertex in A are external to the set A. That is, we have

$$|\partial A| \leq |N(A)| \leq |\partial A| + |A|.$$

Another way to view this is to recognize that the boundary of A is just the neighborhood of A with all the included vertices of A removed. That is, in set notation, we have $\partial A = N(A)-A$.

DEFINITION 4.3.2: An **(n,d,c)-magnifier** is a graph $G=G(V,E)$ on $|V|=n$ vertices, such that for every subset $A\subset V$ where $|A|\leq n/2$, we have

$$|\partial A| = |N(A)-A| \geq c|A|. \qquad \square$$

Hence, for any collection of up to half the vertices, the size of the set of "new neighbors" of that collection is at least the constant c times the size of that collection. Then any such subset A would "magnify" its size by $(1+c)$ when incorporating all its immediate new neighbors. That is, suppose we begin with a subset $A \subset V$ where $|A|\leq n/2$. Then all vertices in the boundary (one edge away), plus the original set, comprise at least (1+c)|A| vertices, two edges away is at least $(1+c)^2|A|$ vertices, etc., until it is clear that there are at least $(1+c)^k|A|$ vertices within distance k of the original set -- valid up until such a growing set is more than half of the entire set V of vertices. Clearly, if a graph is an (n,d,c)-magnifier with a large value of c, then it should have a relatively small diameter since it "expands" well to meet many new neighbors.

DEFINITION 4.3.3: An **(n,d,ε)-enlarger** is a graph on n vertices with maximal degree d and $\lambda_1 \geq \varepsilon$, where λ_1 is the smallest nonzero eigenvalue of the standard matrix Q. □

Clearly, if we can find λ_1 for a given graph G, then it is by definition an (n,d,λ_1)-enlarger. Hence, if we establish a relationship between enlargers and magnifiers, we will have once again related λ_1 to the expansion (magnification) characteristics of the graph. The following two theorems accomplish just that, the first stating that every connected k-regular enlarger is a magnifier for suitable c, the second demonstrating that every magnifier is an enlarger for suitable ε.

THEOREM 4.3.4 ([A], Corollary 2.3): Let $G=(V,E)$, $|V|=n$, be a connected k-regular graph. Then G is an (n,k,c)-magnifier, where c is at least

$$c = 2\lambda_1/(k+2\lambda_1).$$

***Proof*:** By definition, an (n,d,c)-enlarger is a graph on n vertices with maximal degree d and $\lambda_1 \geq c$, where λ_1 is the smallest nonzero eigenvalue of the matrix Q. Clearly, then, G is an (n,k,λ_1)-enlarger, since we trivially have $\lambda_1 \geq \lambda_1$ and d=k.

Let $A \subset V$ be a subset of V such that $|A| \leq n/2$.

Define $B = V-(A \cup N(A))$, where N(A) is the neighborhood of A from Definition 4.3.1.

CASE 1: $B=\emptyset$.

Then $|\partial A| = |N(A)-A| \geq n/2 \geq |A|$, implying that, for this case, $c=1 > 2\lambda_1/(k+2\lambda_1)$ will work.

CASE 2: $|B|\geq 1$.

By definition, B contains only vertices which are not adjacent to A, meaning that any vertex in B is at least two edges away from any vertex in A, and so $\rho\geq 2$ where $\rho=\rho(A,B)$, the length of the shortest path from set A to set B.

Define $a=|A|/n$ and $b=|B|/n$.

Then

$$\begin{aligned}1 - (|A|+|N(A)-A|)/n &= |V|/n - (|A|+|N(A)-A|)/n \\ &= [|V|-(|A|+|N(A)-A|)]/n \\ &= |B|/n \\ &= b \\ &\leq (1-a)/[1 + (\lambda_1/k)a\rho^2], \text{ by Thm. 3.2.2,} \\ &= (1-|A|/n)/[1 + (\lambda_1/k)(|A|/n)2^2] \\ &= (1-|A|/n)/[1 + 4\lambda_1|A|/(kn)].\end{aligned}$$

Hence

$$|N(A)-A|/n \geq [1-|A|/n] - (1-|A|/n)/[1 + 4\lambda_1|A|/(kn)]$$
$$= (1-|A|/n)[1 - 1/\{1 + 4\lambda_1|A|/(kn)\}].$$

Multiplying through by n yields

$$|N(A)-A| \geq (1-|A|/n)[n - n/\{1 + 4\lambda_1|A|/(kn)\}]$$
$$= (1-|A|/n)[n\{1 + 4\lambda_1|A|/(kn)\} - n]/\{1 + 4\lambda_1|A|/(kn)\}$$
$$= (1-|A|/n)[n + (4\lambda_1|A|/k) - n]/\{1 + 4\lambda_1|A|/(kn)\}$$
$$= (1-|A|/n)[4\lambda_1|A|/k]/\{1 + 4\lambda_1|A|/(kn)\}$$
$$= (1-|A|/n)\cdot|A|\cdot(4\lambda_1)/\{k + 4\lambda_1|A|/n\}$$
$$= [(1-|A|/n)(4\lambda_1)/\{k + 4\lambda_1|A|/n\}]|A|.$$

Since we have $|A|\leq n/2$, we know that $(1-|A|/n) \geq ½$ and $|A|/n\leq ½$, so

$$|N(A)-A| \geq [½(4\lambda_1)/\{k+½\cdot 4\lambda_1\}]\cdot|A|$$
$$= [2\lambda_1/(k+2\lambda_1)]\cdot|A|.$$

But $A\subseteq V$ was randomly chosen with the restriction that $|A|\leq n/2$. So, by definition, G is an (n,k,c)-magnifier, where c is at least

$$c = 2\lambda_1/(k+2\lambda_1). \qquad \square$$

Note that this result may be adapted for use with the value of the spectral radius of G. As established in Chapter 2.4, we have that $\lambda_1 \geq k-\mu$. Then we have that

$$\begin{aligned} c &= 2\lambda_1/(k+2\lambda_1) \\ &= 2/[(k/\lambda_1)+2] \\ &\geq 2/[(k/(k-\mu))+2] \\ &= 2(k-\mu)/[k + 2(k-\mu)] \\ &= (2k-2\mu)/(3k-2\mu). \end{aligned}$$

Clearly, Theorem 4.3.4 establishes that, for a k-regular graph, knowing λ_1 provides a minimum value for the constant c: for fixed k, as λ_1 increases, the minimum value of c increases. Since all Cayley graphs are k-regular, Theorem 4.3.4 applies, showing that comparing λ_1's among proposed network models gives some idea as to their relative expansion characteristics.

Recalling from Theorem 3.2.5 that $c_{max} \geq h(G)/d$ and $h(G) \geq \frac{1}{2}\lambda_1$, we already showed that $c_{max} \geq \lambda_1/2d$, where d is the maximum degree of the underlying graph G. It is tempting to assert that the result from Theorem 4.3.4 is superior. However, the allowed range of the subset $A \subset V$ is different when defining c_{max} from the range used to define c. This prevents us from easily drawing any conclusions as to their relative sizes, and therefore limits comparison between the two theorems. In addition, theorem 3.2.5 is valid for all connected graphs with maximum degree d, while Theorem 4.3.4 holds only for k-regular graphs.

In short, we have only succeeded in demonstrating once again that λ_1 is directly related to the expansion characteristics of the underlying graph. The choice between using Theorem 3.2.5 and Theorem 4.3.4 to evaluate the expansion of graphs is thus dependent upon the specific nature of the constants of interest. For the sake of closure, we present essentially the converse of Theorem 4.3.4, which provides a lower bound for λ_1 in terms of c.

THEOREM 4.3.5 ([A], Lemma 2.4): Let $G=(V,E)$ be an (n,d,c)-magnifier. Then G is an (n,d,ε)-enlarger, where $\varepsilon = c^2/(4+2c^2)$. I.e., $\lambda_1 = \lambda_1(G) \geq c^2/(4+2c^2)$.

***Proof*:** Let $f\colon V \mapsto \mathbb{R}$ be an eigenvector of Q corresponding to λ_1. That is, f is an n-tuple with real entries of the form $f = (f(v_1),f(v_2),\ldots,f(v_n))$. From work done in Chapter 2, we know that the constant functions (n-tuples) are eigenvectors of $\lambda_0=0$, and that any nonconstant eigenvectors of Q are orthogonal to a constant function. Since f is associated with $\lambda_1 > 0$, we know f is a nonconstant eigenvector, and so

$$\begin{aligned} 0 &= f\bullet(1,1,\ \ldots\ ,1) \\ &= (f(v_1),f(v_2),\ldots,f(v_n))\bullet(1,1,\ldots,1) \\ &= \sum f(v_i). \end{aligned}$$

Since f is nonconstant, it cannot be the zero function, so in order for the sum of its entries to be zero, some portion of the entries must therefore be positive and some negative. Without loss of generality, we may assume that no more than half are positive. (If not, we may simply choose to use the eigenvector -f for the following argument. Clearly, if f is an eigenvector of λ_1, so is -f, so this assumption is valid.)

Set $V^+ = \{v \in V:\ f(v) > 0\}$ and $V^- = V - V^+ = \{v \in V:\ f(v) \leq 0\}$.

Define with standard notation

$$E(V^+, V^+) = \{(u,v) \in E:\ u \in V^+,\ v \in V^+\}$$

and

$$E(V^+, V^-) = \{(u,v) \in E:\ u \in V^+,\ v \in V^-\}.$$

Then $E(V^+, V^+)$ is the set of edges internal to the set V^+, while $E(V^+, V^-)$ is the bridge of sets V^+ and V^-.

Finally, define a new function (n-tuple) $g: V \mapsto \mathbb{R}$ by

$$g(v) = \begin{cases} f(v) & \text{if } v \in V^+ \\ 0 & \text{otherwise.} \end{cases}$$

Since $Qf = \lambda_1 f = \lambda_1(f(v_1), \ldots, f(v_n))$, we have that

$$
\begin{aligned}
Qf \bullet f &= \lambda_1 (f(v_1), \ldots, f(v_n)) \bullet (f(v_1), \ldots, f(v_n)) \\
&= \lambda_1 \sum f^2(v_i)
\end{aligned}
$$

and hence

$$\lambda_1 = [\sum_{v \in V} (Qf)(v) \cdot f(v)] / [\sum_{v \in V} f(v) \cdot f(v)].$$

But $Qf(v_i) = \lambda_1 f(v_i) = \lambda_1 g(v_i)$ for every $v_i \in V^+ \subseteq V$, so, clearly, if we restrict the above equality to count only vertices $v \in V^+$, the equation is still valid, and we have

$$\lambda_1 = [\sum_{v \in V^+} (Qf)(v) \cdot f(v)] / [\sum_{v \in V^+} f(v) \cdot f(v)].$$

Recalling that $Q = K - A(G)$ where K is the diagonal matrix with $K_{ii} = \deg(v_i)$ and $A(G)$ is the standard adjacency matrix for G, we have

$$
\begin{aligned}
\sum_{v \in V^+} (Qf)(v) \cdot f(v) &= \sum_{v \in V^+} [([K - A(G)] f)(v) \cdot f(v)], \\
&= \sum_{v \in V^+} [(Kf)(v) \cdot f(v) \;-\; (A(G) f)(v) \cdot f(v)] \\
&= \sum_{v \in V^+} [\deg(v) \cdot f^2(v) \;-\; (A(G) f)(v) \cdot f(v)].
\end{aligned}
$$

Since $A(G)$ is the adjacency matrix for G, the i^{th} entry of $A(G)f = (A(G)f)(v_i)$ is the sum of the entries of f whose positions correspond to the subscript of vertices adjacent to v_i in G. That is, $A(G)f$ is an n-tuple which is of the form

$$A(G)f = (\sum_{u \text{ adj to } v_1} f(u), \sum_{u \text{ adj to } v_2} f(u), \ldots, \sum_{u \text{ adj to } v_n} f(u)).$$

Recalling the definition of the neighborhood N(A) of a set A of vertices, when the set is the single vertex v we have

$$N(v) = \{u \in V: (u,v) \in E\},$$

which is simply all the vertices adjacent to v in G. Then we have that

$$\sum_{v \in V^+} (Qf)(v) \cdot f(v) = \sum_{v \in V^+} [\deg(v) f^2(v) - \sum_{u \in N(v)} f(u) \cdot f(v)]. \qquad (*)$$

Consider what this summation looks like when broken down into sections, one for each $v_i \in V^+$. Each such v_i contributes

$$\deg(v_i) f^2(v_i) - [f(v_{1i}) + f(v_{2i}) + \ldots + f(v_{\deg(v_i)i})] f(v_i),$$

where v_{ji} is a vertex of V adjacent in G to v_i. (There are $\deg(v_i)$ of them.)

Then we claim that the following expression, which sums edges in $E(V^+,V^+)$ and $E(V^+,V^-)$, yields the same total as the right-hand side of equation (*) above:

$$\sum_{(u,v)\in E(V^+,V^+)} (f(u)-f(v))^2 + \sum_{(u,v)\in E(V^+,V^-)} f(u)\cdot(f(u)-f(v)).$$

Clearly, for each edge $e=(u,v) \in E(V^+,V^+)$ we get

$$f^2(u)-2f(u)f(v)+f^2(v) = [f^2(u)-f(u)f(v)] + [f^2(v)-f(v)f(u)]$$

while for each edge $e=(u,v) \in E(V^+,V^-)$ we get

$$f^2(u) - f(u)f(v).$$

Each vertex $v_i \in V^+$ will appear $\deg(v_i)$ times in counting these edges, since

$$\deg(v_i) = (\text{\# of verts in } V^+ \text{ adj to } v_i)$$
$$+ (\text{\# of verts in } V^- \text{ adj to } v_i)$$

$$= (\#\text{ of edges in } E(V^+,V^+) \text{ containing } v_i)$$
$$+ (\#\text{ of edges in } E(V^+,V^-) \text{ containing } v_i).$$

Hence, it is clear that each vertex $v_i \in V^+$ has its contributions matched in the two expressions, so our claim is true. That is

$$\sum_{v\in V^+} (Qf)(v)\cdot f(v) = \sum_{v\in V^+} [\deg(v) f^2(v) - \sum_{u\in N(v)} f(u)\cdot f(v)]$$
$$= \sum_{(u,v)\in B(V^+,V^+)} (f(u)-f(v))^2$$
$$+ \sum_{(u,v)\in B(V^+,V^-)} f(u)[f(u)-f(v)].$$

Now, consider the expression $\sum_{(u,v)\in B} [g(u)-g(v)]^2$, taken over all edges in E. For any edge $(u,v) \in E(V^+,V^+)$ we have

$$[g(u)-g(v)]^2 = [f(u)-f(v)]^2,$$

since $g(v)=f(v)$ for every $v\in V^+$, while for any edge $(u,v) \in E(V^+,V^-)$ we have

$$[g(u)-g(v)]^2 = [g(u)-0]^2, \text{ since } g(v)=0 \text{ for } v\in V^-,$$

$$= g^2(u)$$

$$= f^2(u)$$

$$\leq f^2(u)-f(u)f(v),$$

since $f(v) \leq 0$ for all $v \in V^-$ and $f(u) > 0$ for $u \in V^+$.

Finally, for any edge $(u,v) \in E(V^-,V^-)$

$$[g(u)-g(v)]^2 = [0-0]^2 = 0.$$

Since this covers all the edges of E, we have that

$$\sum_{(u,v)\in E(V^+,V^+)} (f(u)-f(v))^2 + \sum_{(u,v)\in E(V^+,V^-)} f(u)[f(u)-f(v)] \geq \sum_{(u,v)\in E} [g(u)-g(v)]^2,$$

and so

$$\sum_{v\in V^+} (Qf)(v)\cdot f(v) \geq \sum_{(u,v)\in E} [g(u)-g(v)]^2.$$

Clearly $\sum_{v\in V^+} f^2(v) = \sum_{v\in V^+} g^2(v) = \sum_{v\in V} g^2(v)$. So, recalling that

$$\lambda_1 = [\sum_{v\in V^+} (Qf)(v)\cdot f(v)]/[\sum_{v\in V^+} f(v)\cdot f(v)],$$

we have by substitution of the above previous two equations that

$$\lambda_1 \geq \Big(\sum_{(u,v)\in E} [g(u)-g(v)]^2\Big)/\Big(\sum_{v\in V} g^2(v)\Big). \qquad (1)$$

We now use the Max-flow Min-cut theorem from graph theory (see [CL], Theorem 5.15) to show that the magnifying properties of G supply a lower bound for the right-hand side of inequality (1). Consider the network N(G) with the vertex set $\{s,t\}\cup X\cup Y$, where s is the source, t is the sink, and $X=V^+$ and $Y=V$ are disjoint sets of vertices. (A sample such network is shown in Figure 4.3.6.) Construct the arcs of N(G) as follows:

(a) For every $u\in X$, the arc (s,u) is assigned capacity 1+c.

(b) For every $u\in X$ and $v\in Y$, the arc (u,v) is assigned capacity 1 if $(u,v)\in E$ or u=v; capacity 0 otherwise. (Note that Alon omits to make mention of assigning arcs with a capacity of 1 if u=v. However, this is required in order to validate his claim, as discussed in Appendix F.)

(c) For every $v\in Y$, the arc (v,t) is assigned capacity 1.

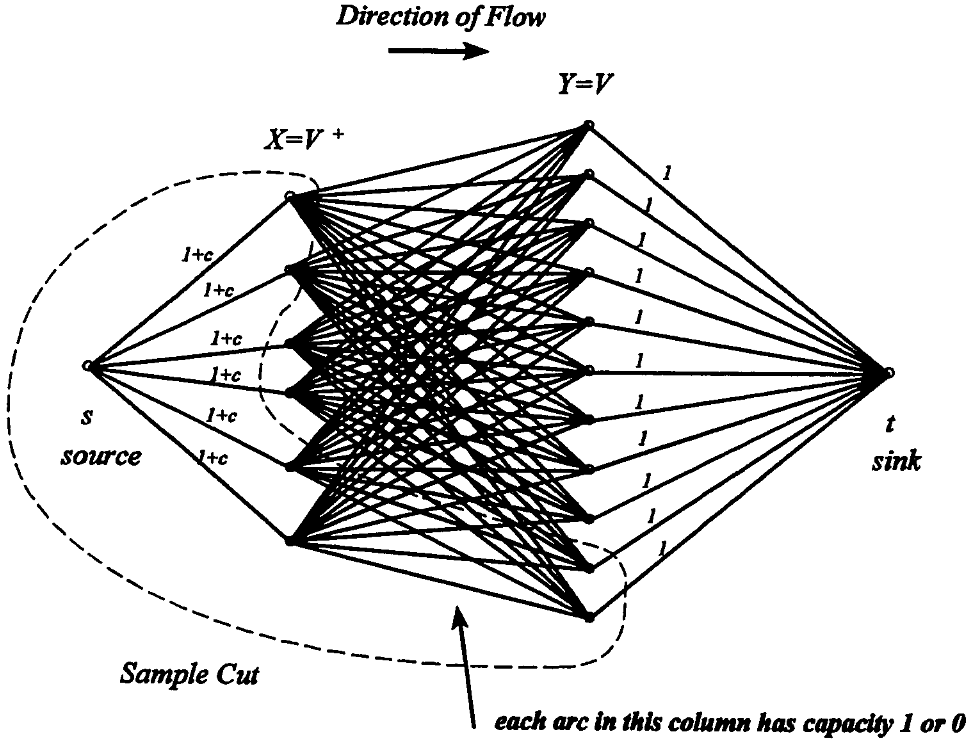

Figure 4.3.6: Sample network with capacities labeled for each arc. Flow is from left (source side) to right (sink side). Note that an arc from the set X to the set Y will be assigned capacity 1 if its end vertices are joined by an edge in the underlying graph or if its end vertices are the same vertex in the underlying graph. Otherwise, such an arc is assigned a capacity of zero. Note the dashed line representing a possible sample cut in the network flow.

Claim: the value of the min-cut of N(G) is $(1+c)|V^+|$.

Clearly, the cut consisting of the partition of vertices into {s} and {all other vertices} leads to a cut with this capacity, since this consists of all arcs (s,u) for every $u \in X$, with each such arc having capacity 1+c and $|X|=|V^+|$. For any other cut C(G), the network is partitioned into the sets

$$\text{left side} = L = \{s \cup U \cup T\}$$

and

$$\text{right side} = R = \{(X/U) \cup (Y/T) \cup t\}$$

where $U = \{u \in V^+ \colon C(G) \text{ does not contain the arc } (s,u)\}$ and T is some (possibly empty) subset of Y.

From network construction, each arc from s to a vertex in X has capacity 1+c. Hence, cut C(G) contains $|X|-|U|$ arcs each with capacity 1+c yielding a subtotal of capacity $(1+c)(|X|-|U|)$.

Due to the magnifying properties of graph G, the subset $U \subseteq X$ has (possibly multiple) arcs of capacity 1 to at least $c|U|$ "new neighbor" vertices (thus distinct from any "subscript partner" vertices) in Y as well as having arcs of capacity 1 to $|U|$ "subscript partners" in Y by the construction parameter (b). Hence, there are $k \geq (1+c)|U|$ vertices in Y that are endpoints of arcs of capacity 1 from one or more vertices in U. Considering each $y \epsilon Y$ of these k vertices, there are two possible scenarios:

(1) $y \epsilon L$, in which case the arc (y,t) is in cut $C(G)$, thereby contributing capacity 1 to cut $C(G)$.

(2) $y \epsilon R$, in which case each arc (u,y) is in cut $C(G)$ where $u \epsilon U$. Each such arc has a capacity of 1, so that each such $y \epsilon R$ contributes at least capacity 1 to cut $C(G)$.

Clearly, in either scenario, each of these k vertices contributes a capacity of at least 1 to cut $C(G)$.

The result of this examination is that cut $C(G)$ has capacity

$$\begin{aligned} \mathrm{Cap}[C(G)] &\geq (1+c)(|X|-|U|) + k(1) \\ &\geq (1+c)(|X|-|U|) + (1+c)|U| \\ &= (1+c)|X| \end{aligned}$$

$$=(1+c)|V^+|.$$

Hence, the claim that the min-cut is $(1+c)|V^+|$ is verified.

Then, by the Max-flow Min-cut Theorem, we can assign a flow to the network equal to $(1+c)|V^+|$ such that it fits the constraints of the capacities. That is, there exists a flow function

$$h:E' \mapsto \mathbb{R},$$

where E' is the edge set of an orientation G' of G (with the addition of a loop at each vertex), such that

(i) flow is between 0 and 1 for each directed arc of G' (0 when capacity is assigned as 0).

(ii) by conservation of flow through all non-source, non-sink vertices, we know that

a) the sum of outflow from each $v \in V^+$ is $1+c$. (Including the flow from v directly to the sink, where capacity is 1, so $h(v,t) \leq 1$.)

b) the sum of the inflow to each $v \in V^+$ is $1+c$, since each flow from the source must be at maximum capacity in order to achieve max flow of $(1+c)|V^+|$.

c) the sum of the outflow from each $v \notin V^+$ is ≤ 1 (else it would exceed capacity from v to the sink).

d) the sum of the inflow to each $v \notin V^+$ is ≤ 1 (since it must equal its outflow).

Then, the assignments of flow functions to the actual edges of the graph G are listed below:

$u \in V^+$: $\sum h(u,v) = 1+c$, where outflow counts flow from u to other vertices in G. (Note that within the set V^+ flows between such vertices are assigned to 0, except for a flow of 1 being assigned from each vertex to itself as a loop.) The summation is taken over all possible v's, including u itself.

$\sum h(v,u) = 1$, where the inflow is 1 only because we assign a flow of 1 from u to itself. The rest are all set to 0.

$u \notin V^+$: $\sum h(u,v) = 0$, so that each of these outflows is assigned a value of 0. (All loops are set to zero. Note that the only flow out from these vertices in the constructed network is out to the sink, which is not a vertex of the graph, so no such edges exist in G.)

$\sum h(v,u) \leq 1$, since all inflows must sum to ≤ 1. (These are the flows from the $u \in V^+$. All loops are set to zero.)

Note that each summation is taken over all possible v's, including u itself. Observe also that these are all natural constraints of the network constructed in Alon, with the additional property that capacity of arc $(u,u)=1$ for every $u \in V^+$.

Combining this function with our previously defined function $g: V \mapsto \mathbb{R}$, we find that the following two claims are valid:

Claim (1): $\sum_{(u,v)\in B'} h^2(u,v)\,[g(u)+g(v)]^2 \leq 2(2+c^2) \sum_{u\in V} g^2(u)$.

proof:

$$\sum_{(u,v)\in B'} h^2(u,v)\,[g(u)+g(v)]^2$$

$$= \sum_{(u,v)\in B'} h^2(u,v)\,[g^2(u)+2g(u)g(v)+g^2(v)]$$

$$\leq \sum_{(u,v)\in B'} h^2(u,v)\,[2g^2(u)+2g^2(v)],$$

since $2a^2+2b^2 \geq a^2+2ab+b^2$

$\Leftrightarrow \; a^2+b^2 \geq 2ab$

$\Leftrightarrow \; a^2-2ab+b^2 \geq 0$

$\Leftrightarrow \; (a-b)^2 \geq 0$, which is true,

$$= 2\sum_{(u,v)\in B'} h^2(u,v)\,[g^2(u)+g^2(v)]$$

$$= 2\sum_{(u,v)\in B'} h^2(u,v)g^2(u) + 2\sum_{(u,v)\in B'} h^2(u,v)g^2(v)$$

$$= 2\sum_{(u,v)\in B'} h^2(u,v)g^2(u) + 2\sum_{(u,v)\in B'} h^2(v,u)g^2(u),$$

by an index change,

$$= 2\sum_{u\in V} g^2(u)\,[\sum_{v\in V;\,(u,v)\in B'} h^2(u,v) + \sum_{v\in V;\,(v,u)\in B'} h^2(v,u)].$$

This last step is done by factoring out the $g^2(u)$ from each term and summing by considering each $u\in V$, and then considering each oriented edge affiliated with that particular u for the subtotals. Clearly, each oriented edge contributes just as in the line above. Finally, when considering each $u \in V$, if $u \notin V^+$, then $g(u)=0$, so that the only parts of the last line which contribute nonzero terms are when $u \in V^+$. Evaluating the first term for a particular $u \in V^+$ we have

$$\sum_{v \in V;\, (u,v) \in B'} h^2(u,v) = h^2(u,u) + \sum_{v \in V-\{u\};\, (u,v) \in B'} h^2(u,v)$$
$$= 1^2 + \sum_{v \in V-\{u\};\, (u,v) \in B'} h^2(u,v),$$

since $h(u,u)=1$. But we know that the total outflow assigned to any vertex $u \in V^+$ is $1+c$, so that the remaining vertices in $V\{u\}$ can receive at most a flow of total c from vertex u, since $h(u,u)=1$. Since all flows are positive, we have that the summation excluding $h^2(u,u)$ is

$$\sum h^2(u,v) \leq (\sum h(u,v))^2 = c^2.$$

Thus

$$\sum_{v \in V-\{u\};\, (u,v) \in B'} h^2(u,v) \leq 1+c^2 \text{ for each } u \in V^+.$$

Evaluating the second term, we get

$$\sum_{v\in V;\,(v,u)\in B'} h^2(v,u) \;=\; 1^2 + 0 = 1$$

since, as noted above, the only inflow to a vertex $u\in V^+$ is from itself, so that $1 = h(u,u) = h^2(u,u)$ while all other values of $h(v,u)$ are set to zero when $v\neq 0$. Thus, for each particular $u\in V^+$ we have that

$$\Big[\sum_{v\in V;\,(u,v)\in B'} h^2(u,v) \;+\; \sum_{v\in V;\,(v,u)\in B'} h^2(v,u)\Big] \;\le\; 1+c^2+1 \;=\; 2+c^2.$$

Hence, when summing over all $u\in V^+$, we have that

$$\sum_{(u,v)\in B'} h^2(u,v)\,[g(u)+g(v)]^2 \;\le\; 2(2+c^2)\sum_{u\in V} g^2(u),$$

and claim (1) is verified. (□)

Claim (2): $c = \{\sum_{(u,v)\in B'} h(u,v)\,[g^2(u)-g^2(v)]\}/\sum_{u\in V} g^2(u)$.

proof:

$$\sum_{(u,v)\in B'} h(u,v)\,[g^2(u)-g^2(v)]$$

$$= \sum_{(u,v)\in B'} h(u,v)g^2(u) \;-\; \sum_{(u,v)\in B'} h(u,v)g^2(v)$$

$$= \sum_{(u,v)\in B'} h(u,v)\,g^2(u) - \sum_{(u,v)\in B'} h(v,u)\,g^2(u),$$

by a change of index,

$$= \sum_{u\in V} g^2(u)\,[\sum_{v\in V;\,(u,v)\in B'} h(u,v) - \sum_{v\in V;\,(v,u)\in B'} h(v,u)],$$

by again considering these sums while working through each $u\in V$. As before, since $g(u)=0$ for any $u\notin V^+$, we need only consider the nonzero terms in the summation over all $u\in V^+$, so

$$\sum_{(u,v)\in B'} h(u,v)\,[g^2(u)-g^2(v)]$$

$$= \sum_{u\in V^+} g^2(u)\,[\sum_{v\in V;\,(u,v)\in B'} h(u,v) - \sum_{v\in V;\,(v,u)\in B'} h(v,u)].$$

The first term in the parenthesis, evaluated for a particular $u\in V^+$, yields

$$\sum_{v\in V;\,(u,v)\in B'} h(u,v) = 1+c,$$

by the same reasoning as above, while a similar evaluation of the second term yields

$$\sum_{v\in V;\,(v,u)\in B'} h(v,u) = 1,$$

since all flows into u are zero except for h(u,u)=1. Thus, for every $u \in V^+$

$$\sum_{v\in V;\,(u,v)\in B'} h(u,v) \;-\; \sum_{v\in V;\,(v,u)\in B'} h(v,u) \;=\; (1+c)-1 \;=\; c,$$

and so

$$\sum_{(u,v)\in B'} h(u,v)\,[g^2(u)-g^2(v)] \;=\; c\sum_{u\in V} g^2(u)\,.$$

Hence

$$c \;=\; \Big\{ \sum_{(u,v)\in B'} h(u,v)\,[g^2(u)-g^2(v)] \Big\} \Big/ \sum_{u\in V} g^2(u)\,,$$

and claim (2) is verified. (□)

Recall that

$$\lambda_1 \;\geq\; \Big(\sum_{(u,v)\in B} [g(u)-g(v)]^2 \Big) \Big/ \Big(\sum_{v\in V} g^2(v) \Big)$$

$$= \frac{\Big(\sum_{u\in V^+} [g(u)-g(v)]^2 \Big) \Big(\sum_{(u,v)\in B'} h^2(u,v)\,[g(u)+g(v)]^2 \Big)}{\Big(\sum_{v\in V} g^2(v) \Big) \Big(\sum_{(u,v)\in B'} h^2(u,v)\,[g(u)+g(v)]^2}$$

which is simply from multiplying by one. Considering the numerator, we see that the first factor sum may be viewed as the dot product of a vector with itself (i.e., the square of the norm) where that vector has |E| entries. That is

$$\sum_{u \in V^*} [g(u)-g(v)]^2 = x \bullet x = \|x\|^2,$$

where $x = (x_1, x_2, \ldots, x_{|E|})$ and $x_i = g(u)-g(v)$, where u and v are the end vertices of the ith edge in E. Similarly, we may view the second factor sum of the numerator as a dot product of a vector with itself where the vector has |E'|=|E| entries. That is

$$\begin{aligned}\sum_{(u,v) \in E'} h^2(u,v)[g(u)+g(v)]^2 &= \sum_{(u,v) \in E'} \{h(u,v)[g(u)+g(v)]\}^2 \\ &= y \bullet y \\ &= \|y\|^2\end{aligned}$$

where $y = (y_1, y_2, \ldots, y_{|E|})$ and $y_i = h(u,v)[g(u)+g(v)]$, where u and v are the end vertices of the i^{th} edge in E'. Then we have for the numerator that

$$\sum_{u \in V^*} [g(u)-g(v)]^2 \sum_{(u,v) \in E'} h^2(u,v)[g(u)+g(v)]^2$$
$$= \|x\|^2 \|y\|^2$$

$$\begin{aligned}
&\geq |x \bullet y|^2, \text{ by Cauchy-Schwarz,} \\
&= | \sum_{(u,v)\in B'} (x_i y_i) |^2 \\
&= | \sum_{(u,v)\in B'} [g(u)-g(v)]h(u,v)[g(u)+g(v)] |^2 \\
&= | \sum_{(u,v)\in B'} h(u,v)[g^2(u)-g^2(v)] |^2 \\
&= (\sum_{(u,v)\in B'} h(u,v)[g^2(u)-g^2(v)])^2 .
\end{aligned}$$

Recalling from inequality (1) above, that

$$\sum_{(u,v)\in B'} h^2(u,v)[g(u)+g(v)]^2 \leq 2(2+c^2) \sum_{u\in V} g^2(u)$$

we have directly for the denominator

$$\begin{aligned}
&(\sum_{v\in V} g^2(v)) \sum_{(u,v)\in B'} h^2(u,v)[g(u)+g(v)]^2 \\
&\qquad \leq (\sum_{v\in V} g^2(v))[2(2+c^2)\sum_{u\in V} g^2(u)] \\
&\qquad = 2(2+c^2)[g^2(v)]^2 .
\end{aligned}$$

Then we may finally evaluate λ_1 as

$$\lambda_1 \geq \frac{(\sum_{u\in V^*} [g(u)-g(v)]^2)(\sum_{(u,v)\in B'} h^2(u,v)[g(u)+g(v)]^2)}{(\sum_{v\in V} g^2(v))(\sum_{(u,v)\in B'} h^2(u,v)[g(u)+g(v)]^2}$$

$$\geq \Big(\sum_{(u,v)\in B'} h(u,v)\,[g^2(u)-g^2(v)]\Big)^2 / \{2(2+c^2)\,[\sum_{v\in V} g^2(v)]^2\}$$

$$= [1/(4+2c^2)]\{\Big(\sum_{(u,v)\in B'} h(u,v)\,[g^2(u)-g^2(v)]\Big)^2 / [\sum_{v\in V} g^2(v)]^2\}$$

$$= [1/(4+2c^2)]\{\Big[\sum_{(u,v)\in B'} h(u,v)\Big]\,[g^2(u)-g^2(v)] / [\sum_{v\in V} g^2(v)]\}^2$$

$\geq [1/(4+2c^2)]c^2$, by statement (2) above.

Hence

$$\lambda_1 \geq c^2/(4+2c^2). \qquad \square$$

It is of interest to note that this inequality may be manipulated to provide an upper bound for c in terms of λ_1, but that this boundary is no longer valid for $\lambda_1 \geq ½$, and is quite sensitive when λ_1 is near ½. (Clearly, $c^2/(4+2c^2)$ can never exceed ½, regardless of how large c may be.) For this reason, the inequality $(8d\lambda_1)^{½} \geq c_{max}$ provided by Theorem 3.2.5 is a preferred form for finding the upper bound on an expansion-related constant.

4.4 Families of Ramanujan Graphs

In [A] and [AM], the results of Theorems 4.3.4 and 4.3.5 are used as the conceptual building blocks for techniques in the construction of desirable expanders, superconcentrators, and infinite families of linear enlargers. In particular, [AM] introduce the concept of a group having **property (T)**, which is similar to the concept of a group being Kazhdan, as examined in Chapter 3. Their emphasis is placed on producing families of graphs with "verifiably good" expanding properties.

In [Sch], [LPS], and [Ch], the initial focus of discussion about Theorem 4.3.4 is on obtaining some sense of the limits of expansion capabilities of graphs and, in particular, infinite families of graphs. As stated in [Sch], "the question arises how small μ can be made" (or, equivalently, how large can λ_1 be made) for an infinite family of k-regular graphs? Such a focus led to establishing the benchmark Ramanujan graphs.

Subsequently, these authors attempt to produce graphs of equal or superior expansion to those of Ramanujan graphs.

We present here the theorem leading to the Ramanujan bounds, and will, in Chapter 6, examine the family of Cayley graphs on Z_n for Ramanujan characteristics.

THEOREM 4.4.1 ([LPS], Proposition 4.2): Let $\{G_n\}$ be any family of k-regular (connected) graphs on n vertices, where k is fixed and n takes on infinitely many values in N. Then

$$\lim_{n\to\infty} \inf\{\mu(G_n)\} \geq 2(k-1)^{½}.$$

Proof: Let A(G) be the standard adjacency matrix of G=G(V,E) for |V|=n.

Let $A(G)^t$ denote the t^{th} power of the adjacency matrix. Using the notation that $\delta_{ij}^{(t)}$ is the ij^{th} entry of $A(G)^t$, we recall a straightforward property of incidence, or adjacency, matrices that the value of $\delta_{ij}^{(t)}$ gives the number of walks of length t joining vertex v_i to vertex v_j in graph G. (Note that these entries provide the number of walks (**not** paths!) as conventionally used in the terminology of graph theory, since it counts routes with repeated vertices. For a simple example, consider the adjacency matrix of a graph with two vertices and an edge between them. $A(G)^3$ clearly says that there is a walk of length three from one vertex to the other. In order for this to be possible, each vertex must be repeated.) Recall two properties of the trace of an n×n matrix:

(1) tr(AB) = tr(BA). *proof*:

$$\begin{aligned} tr(AB) &= \sum_{i=1}^{n} [a_{i1}b_{1i}+a_{i2}b_{2i}+\ldots+a_{in}b_{ni}] \\ &= a_{11}b_{11}+\ldots+a_{1n}b_{n1}+a_{21}b_{12}+\ldots+a_{2n}b_{n2}+\ldots+a_{n1}b_{1n}+\ldots+a_{nn}b_{nn} \\ &= (\text{row 1 of B})(\text{col 1 of A})+(\text{row 2 of B})(\text{col 2 of A}) \\ &\qquad +\ldots+(\text{row n of B})(\text{col n of A}) \\ &= tr(BA). \qquad (\square) \end{aligned}$$

(2) The trace of similar matrices are equal. *proof*:

$$\begin{aligned} tr(QAQ^{-1}) &= tr[(QA)Q^{-1}] \\ &= tr[Q^{-1}(QA)] \\ &= tr[(Q^{-1}Q)A] = tr(A). \qquad (\square) \end{aligned}$$

Hence, for any diagonalizable matrix M, we have that

$$\mathrm{tr}(M) = \mathrm{tr}(M\text{'s diagonalized matrix}).$$

Since A(G) is a symmetric real matrix, a standard linear algebra theorem ([FIS], Theorem 6.20) states that it is orthogonally equivalent to a real diagonal matrix (ensuring that it has a full complement of eigenvalues). That is, we can find an orthogonal matrix Q of eigenvectors of A(G) and a diagonal matrix D consisting of the eigenvalues of A(G) such that $Q^{-1}A(G)Q = D$. Thus

$$\mathrm{tr}(A(G)) = \mathrm{tr}(D)$$

where D is the diagonal matrix with its entries the eigenvalues of A(G), since D is similar to A(G). That is

$$\sum_{i=1}^{n} \delta_{ii} = \sum_{i=1}^{n} \xi_i,$$

where $k=\xi_1 \geq |\xi_2| \geq \ldots \geq |\xi_n|$ are the eigenvalues of A(G).

Consider higher powers of A(G) such as $A(G)^t$, for $t \geq 2$. Since $A(G)^t$ is still symmetric and therefore remains a diagonalizable matrix, its trace is equal to the trace of its diagonalized (eigenvalue) matrix. That is

$$\sum \text{(eigenvalues of } A(G)^t\text{)} = tr(A(G)^t) = \sum \delta_{ii}{}^{(t)},$$

where $\delta_{ii}{}^{(t)}$ is the iith entry of $A(G)^t$ (**not** the tth power of the iith entry of A(G)). Observe that if ξ is the eigenvalue of A(G) corresponding to f, then

$$A(G)^2 f = A(G)(A(G)f) = A(G)(\xi f) = \xi(A(G)f) = \xi(\xi f) = \xi^2 f.$$

Hence, ξ^2 is an eigenvalue of $A(G)^2$, and, of course, an inductive argument shows that ξ^t is an eigenvalue of $A(G)^t$. Then we have that

$$\sum \xi_i{}^t = \sum \delta_{ii}{}^{(t)}.$$

Since it is simpler to obtain formulas for the number of certain kinds of possible walks in the k-regular tree T^k, it is useful to compare the k-regular tree with the k-regular graph on n vertices. Consider a walk of length t that begins at a particular vertex in T^k and ends back at the same vertex. Such a walk will have a corresponding walk in our k-regular graph. (Note that the issue of t being larger than the diameter of G is irrelevant: for each walk in T^k that extends more than diam(G) from the starting vertex, a cycle in G can "take up the slack". In this way, walks in G can continue on for as large a value of t as desired to match a walk in T^k.)

In addition, the availability of loops in the k-regular Cayley graph G provides additional choices for walks, while the k-regular tree is acyclic (has no loops). Hence, the number of distinct walks of length t from a vertex back to itself in $A(G)^t$ is greater than or equal to the number of such distinct walks in the k-regular tree. That is, for every vertex in G, and thus for every i between 1 and n, we have

$$\delta_{ii}{}^{(t)} \geq \rho(t)$$

where ρ(t) is the number of walks of length t starting and ending at a given vertex in T^k. Thus

$$\sum_{i=1}^{n} \xi_i^t \geq n\rho(t).$$

Subtracting $\xi_1^t = k^t$ from both sides yields

$$\sum_{i=2}^{n} \xi_i^t \geq n\rho(t)-k^t.$$

Subtracting $\xi_2^t \leq k^t$ from both sides yields

$$\sum_{i=3}^{n} \xi_i^t = \left(\sum_{i=2}^{n} \xi_i^t\right) - \xi_2^t \geq n\rho(t)-k^t-k^t = n\rho(t)-2k^t.$$

Recalling from Theorem 4.3.1, part (c), that the eigenvalue k (and also, in the bipartite case, -k) is simple (of multiplicity 1), we have that $\mu \geq |\xi_i|$ for any i between 3 and n, and so clearly also for $\mu^t \geq |\xi_i|^t \geq \xi_i^t$. Then

$$(n-2)\mu^t \geq \sum_{i=3}^{n} \xi_i^t \geq n\rho(t)-2k^t.$$

Since the above equation is true for all t, it is true for all 2t. Substituting yields

$$\begin{aligned}\mu^{2t} &\geq [n\rho(2t)-2k^{2t}]/(n-2)\\ &= (n/(n-2))\rho(2t) - 2k^{2t}/(n-2)\\ &\geq \rho(2t) - 2k^{2t}/(n-2).\end{aligned}$$

Obviously, $\rho(2t) \geq \rho'(2t)$, where $\rho'(2t)$ is the number of walks of length 2t beginning at a vertex and returning to the same vertex for the first time (in T^k). From work on Catalan numbers in Appendix C, we have that

$$\rho'(2t) = (1/t)C(2t-2,t-1)k(k-1)^{t-1},$$

where C(,) is the standard combination notation.

Hence,

$$\begin{aligned}\mu^{2t} &\geq \rho(2t) - 2k^{2t}/(n-2)\\ &\geq \rho'(2t) - 2k^{2t}/(n-2)\\ &= (1/t)C(2t-2,t-1)k(k-1)^{t-1} - 2k^{2t}/(n-2)\\ &\geq (1/t)C(2t-2,t-1)(k-1)(k-1)^{t-1} - 2k^{2t}/(n-2)\end{aligned}$$

$$= (1/t)C(2t-2,t-1)(k-1)^{t} - 2k^{2t}/(n-2)$$

$$= (1/t)C(2t-2,t-1)[(k-1)^{1/2}]^{2t} - 2k^{2t}/(n-2).$$

Considering the limit of this as n tends to infinity, we get

$$\lim_{n\to\infty} \mu^{2t} \geq \lim_{n\to\infty}\{(1/t)C(2t-2,t-1)[(k-1)^{1/2}]^{2t} - 2k^{2t}/(n-2)\}$$

$$= (1/t)C(2t-2,t-1)[(k-1)^{1/2}]^{2t},$$

since $2k^{2t}/(n-2)$ goes to zero as $n\mapsto\infty$ while the rest remains unaffected. Then

$$\lim_{n\to\infty} \mu \geq [(1/t)C(2t-2,t-1)]^{1/(2t)}(k-1)^{1/2}.$$

Since the assumption that the last term is essentially zero is valid for any t that is "significantly smaller" than n, we can see what the trend is for the value of the coefficient of the remaining term for various values of t. The lower bound on μ will then depend on what the largest value of that coefficient can be. We have for the coefficient

$$(1/t)C(2t-2,t-1) = (1/t)[(2t-2)!]/[(t-1)!(t-1)!]$$
$$= [(2t-2)(2t-3)\cdots3\cdot2]/[t(t-1)^2(t-2)^2\cdots3^2\cdot2^2]$$

where the numerator contains 2t-3 factors (including the "central" factor of 2t-t=t), while the denominator contains 2(t-2)+1 = 2t-4+1 = 2t-3 factors as well. After cancelling the factor of t from both top and bottom, we can compare sequential pairs of top factors with sequential pairs of bottom factors, where we have that

$$(2t-2)(2t-3)/(t-1)^2 \mapsto 4 \text{ as } t\mapsto\infty,$$

which approaches 4 from below, since 2t-3 < 2t-2,

$$(2t-4)(2t-5)/(t-2)^2 \mapsto 4 \text{ as } t\mapsto\infty, \text{ (again from below)}$$

and so on.

Hence, we get (t-1)-1 = t-2 such terms in the limit, so that

$$\begin{aligned}\lim_{t\to\infty}[(1/t)C(2t-2,t-1)]^{1/(2t)} &= \lim_{t\to\infty}[(4)^{t-2}]^{1/(2t)} \\ &= \lim_{t\to\infty}(2^{2t-4})^{1/(2t)} \\ &= \lim_{t\to\infty}2^{(2t-4)/(2t)} \\ &= 2,\end{aligned}$$

where the limit approaches 2 from below. Hence, the coefficient is always less than 2, though we can get arbitrarily close with an appropriate value of t. (Note that the coefficient gets "close" rather fast: when t=5, its value is approximately 1.30; when t=10, it is 1.53; when t=20, it is 1.97; and so on.) Therefore, we can say that the limit for μ as n gets very large is

$$\lim_{n\to\infty} \mu \geq 2(k-1)^{1/2}. \qquad \square$$

Thus, we arrive at a "benchmark" of sorts, the lower bound for how small μ may be for infinite families of expanders. Clearly, a smaller μ (or larger λ_1) provides a better lower bound for the expansion coefficient c. As noted before, producing families with high coefficient c is the thrust of much of the literature we surveyed. This bound, therefore, is apparently important enough to name (if, indeed, it was not already named for other reasons).

DEFINITION 4.4.2: a k-regular graph G on n vertices is called a **Ramanujan graph** if

$$\mu(A(G)) \leq 2(k-1)^{1/2}.$$

A **family of Ramanujan graphs** is a sequence of Ramanujan graphs on n_i vertices such that $n_i \mapsto \infty$ as $i \mapsto \infty$. □

Through extensive work with the techniques of number theory, [LPS] construct families of Ramanujan graphs, while [Ch] focuses on families of so-called k-sum and k-difference graphs, which she shows to be good (though not Ramanujan) expanders with small diameters. [Sch], however, attempts to construct individual graphs with substantially smaller spectral radius than the Ramanujan bound. This approach follows the premise that individual graphs (not families) are used in real-world applications, so constructing specific graphs with superior expansion characteristics is a better goal.

Our efforts in Chapter 6 also follow this tactic: though we show that Z_n-based Cayley graphs will not produce Ramanujan families, we do find good expansion characteristics (large λ_1) and small diameters for specific graphs.

CHAPTER 5

SCHIBELL AND STAFFORD'S ROUTING ALGORITHM

5.1 Introduction

Although there exist general algorithms for finding the shortest path from one vertex to another in a graph (e.g., Dijkstra's), the processing power and time required to obtain such paths are prohibitive in parallel-processing networks where the processors are basic and routing time is critical. Current network architectures are thus generally constrained to models which allow extremely efficient, specifically-designed routing algorithms.

Since [SS] wished to propose a different class of groups as the underlying basis for Cayley graph models of parallel-processing networks, they set out to develop a general purpose routing algorithm for such Cayley graphs. This chapter follows the development of their algorithm, and examines its efficiency in several cases.

5.2 Routing in Cayley Graphs Based on Subgroups of S_n

Routing from one vertex to another in a Cayley graph requires finding a path along the graph edges between these two vertices. Suppose the string of edges s_1--s_2--...--s_t represents such a path from vertex x to vertex y in the Cayley graph G. Since movement along an edge corresponds to left multiplication by that edge's generating element in the underlying group Γ (see Chapter 1.2), such a path corresponds to the string of elements

$$x\text{--}s_1x\text{--}s_2s_1x\text{--}\ldots\text{--}s_{t-1}\cdots s_2s_1x\text{--}s_ts_{t-1}\cdots s_2s_1x=y$$

in group Γ.

Clearly, then, $s_t\cdots s_2s_1 = yx^{-1}\epsilon\Gamma$, so any path from x to y may be represented by a "word" of generating elements.

Conversely, any factorization $s_r s_{r-1} \cdots s_2 s_1$ of yx^{-1} into generating elements represents a valid path s_1--s_2--...--s_r from x to y in the Cayley graph.

Hence, if a convenient algorithm for obtaining such a factorization can be found, the problem of routing in the Cayley graph is essentially solved. Furthermore, if a factoring algorithm is valid for an entire class of groups, then the routing for the class of Cayley graphs based on such underlying groups is solved. Specifically, [SS] propose that the Sims factoring algorithm be used to factor permutation groups with generating sets that are "strong with respect to an ordered base". The following definitions provide the meaning to this phrase.

DEFINITION 5.2.1: An **(ordered) base** for a permutation group $\Gamma \subset S_n$ is defined to be an ordered subset $B \subset \{1,2,...,n\}$ such that if σ is a permutation in Γ that sends every $b \in B$ to itself, then σ must be the identity permutation. □

For example, $B=\{3,1\}$ is an ordered base for the permutation group S_3, since any permutation in S_3 that sends 1 to 1 and 3 to 3 must also send 2 to 2. That is, it must be the identity permutation. Note, however, that $B=\{1\}$ is not sufficient to be an ordered base for S_3, since the permutation $(23)\epsilon S_3$ sends 1 to 1, but is not the identity permutation. In essence, an ordered base must contain enough entries so that any permutation is completely determined by where it sends each member of the ordered base.

It is useful here to establish notation for a nested set of subgroups of Γ with respect to a strong base $B=\{b_1,...,b_t\}$. We define the subgroup

$$\Gamma^{k+1} = \{\sigma\epsilon\Gamma: \sigma(b_i)=b_i, \ 1 \le i \le k\}$$

containing every permutation of Γ which fixes the first k elements of the ordered base. Clearly, Γ^{t+1} is merely the identity permutation in Γ, since it is the only permutation to fix every element of the ordered base, by definition. Also, $\Gamma^{k+1} \subset \Gamma^k$ since every permutation in the left-hand set must fix the first k-1 elements of the ordered base in addition to fixing the kth element. Then we clearly have a nested set of subgroups of Γ which stabilizes to the identity in the sequence

$$\Gamma^1=\Gamma \supset \Gamma^2 \supset \Gamma^3 \supset \ldots \supset \Gamma^t \supset \Gamma^{t+1}.$$

Definition 5.2.2: A set of generators S of Γ is said to be a set of **strong generators with respect to the ordered base** $B=\{b_1,b_2,\ldots,b_t\}$ if S contains a set of generators for each of the subgroups in the nested sequence

$$\Gamma \supset \Gamma^2 \supset \Gamma^3 \supset \ldots \supset \Gamma^t. \qquad \square$$

This definition simply ensures that the intersection of any particular nested subset (produced by the ordered base) and the generating set S will contain sufficient elements to generate that nested subset. Note that the definition is not concerned with generating the identity subgroup. It is enough to acknowledge that the identity permutation is the only permutation in Γ which fixes every element in the ordered base. (This exception is necessary since, by definition, the generating set of a Cayley graph does not contain the identity.)

The Cayley graph on $A_4 \subset S_4$ with generating set $S=\{(12)(34),(123),(132)\}$ is an example of a set of strong generators with respect to the ordered base $B=\{4,1\}$. A_4^2 is the cyclic subgroup of A_4 that is generated by the permutation $(123) \in S$, while A_4^3 is the subgroup containing only the identity permutation. Hence, the definition is satisfied.

Recall that $\Gamma^{k+1} \subset \Gamma^k$ for each k from 1 through t. That is, Γ^{k+1} is a subgroup of Γ^k. Then we note that there exists a set of cosets of Γ^{k+1} in Γ^k. Denote U^k as a complete set of coset representatives of Γ^{k+1} in Γ^k. Since the nature of cosets is to partition the main group into subsets of like behavior, we explore that behavior by supposing that the two elements σ and ρ are in the same coset of Γ^{k+1} in Γ^k. Since every element in Γ^k fixes the first k-1 elements of the ordered base B, σ and ρ also fix those first k-1 elements. The question is where each sends b_k. Since they are in the same coset of Γ^{k+1} in Γ^k, we have that $\sigma\Gamma^{k+1}$ and $\rho\Gamma^{k+1}$ are simply different names for the same coset. That is, $\sigma\Gamma^{k+1} = \rho\Gamma^{k+1}$. But this implies that

$$\begin{aligned} & \rho^{-1}\sigma\Gamma^{k+1} = \Gamma^{k+1} \\ \Rightarrow\quad & \rho^{-1}\sigma \in \Gamma^{k+1} \\ \Rightarrow\quad & [\rho^{-1}\sigma](b_k) = b_k \\ \Rightarrow\quad & \sigma(b_k) = \rho(b_k). \end{aligned}$$

Hence, elements from the same coset of Γ^{k+1} in Γ^k send b_k to the same value. Then U^k, the complete set of coset representatives of Γ^{k+1} in Γ^k is a set of permutations, each of which sends b_k to a different value. Since the cosets of Γ^{k+1} in Γ^k partition Γ^k, the set of values where b_k may be sent by the elements of U^k is the complete set of values for all the elements of Γ^k. The result of all this is shown in the following theorem:

THEOREM 5.2.3 (Correction of [SS], Proposition 2.2): Let $\Gamma \subset S_n$ be a permutation group with strong generating set S with respect to the ordered base $B=\{b_1,...,b_t\}$. Let U^i be a complete set of coset representatives of Γ^{i+1} in Γ^i, for every i from 1 to t. Then every element of Γ has a unique representation of the form $U_1 \circ U_2 \circ ... \circ U_{t-1} \circ U_t$, where $U_i \in U^i$.

Proof: Suppose $|B|=t=1$. Then $B=\{b_1\}$, so $\Gamma^2=\{e\}$, the identity permutation. Then $U^1=\Gamma$, since all the elements in Γ are required to name each of the coset representatives of {e}. Then, trivially, any element in Γ may be uniquely written by a word that is an element of U^1, the complete set of coset representatives for {e}. Hence, the claim is true for any permutation group with a strong generating set on an ordered base of cardinality 1.

Let $k \geq 1$ be a positive integer such that the claim is true.

Consider $t=k+1$.

Suppose we have $\sigma \in \Gamma$ such that $\sigma(b_1)=x_j$ for some $x_j \in \{1,...,n\}$. Then, since U^1 is a complete set of coset representatives, σ must share a coset with precisely one of the elements of U^1. That is, there is a unique coset representative $U_{1j} \epsilon U^1$ that also sends b_1 to x_j. Note, then, that $(U_{1j})^{-1}$ sends x_j to b_1. Define the following:

$$\sigma^{(2)} = [(U_{1j})^{-1}] \circ \sigma$$

$$S^{(2)} = S \cap \Gamma^2$$

$$\text{and} \qquad B^{(2)} = B - \{b_1\}.$$

Clearly, $\sigma^{(2)}$ fixes b_1, and so $\sigma^{(2)} \epsilon \Gamma^2$, by definition. In addition, it is now apparent from definitions that Γ^2 is strongly generated with base $B^{(2)}$ and strong generating set $S^{(2)}$. Since the cardinality of base $B^{(2)}$ is $|B-\{b_1\}| = t-1 = k$, by the inductive hypothesis we have a unique representation for $\sigma^{(2)}$ of

$$\sigma^{(2)} = U_2 \circ U_3 \circ \ldots \circ U_t$$

and so, uniquely,

$$\sigma = U_{1j} \circ \sigma^{(2)} = U_{1j} \circ U_2 \circ U_3 \circ \ldots \circ U_t.$$

Hence, the claim is true for t=k+1, and, by induction, for all cardinalities of the ordered base. □

It is appropriate to note here that [SS] present a version of Proposition 2.2 which claims that the representation found is of the form $U_tU_{t-1}\cdots U_2U_1$, instead of the reverse order shown in Theorem 5.2.3 above. During their proof, they set $\sigma^{(2)} = \sigma\circ[U_{1j}]^{-1}$ and claim that $\sigma^{(2)}\epsilon\Gamma^2$. This is simply an incorrect conclusion: it is unknown where $[U_{1j}]^{-1}$ sends the first base point b_1 and it is certainly not guaranteed that it will send b_1 to the value which σ sends back to b_1. The only known quantity about $\sigma\circ[U_{1j}]^{-1}$ is that it fixes x_j. The result of this alteration in the representation found is that [SS] believe that the first edge named by their full algorithm will be the first edge needed to travel the path from vertex x to vertex y. In fact, this is not the case, and an adjustment to their algorithm must be made. This will be discussed in Chapter 5.3.

Clearly, if we can construct an algorithm that produces such coset representatives and factors each $\sigma\epsilon\Gamma$ into a word of those representatives as in Theorem 5.2.3 above, then we have accomplished our goal of factoring each element in Γ.

Since we assume a strong set S of generators with respect to an ordered base, each of the coset names in a given set U^i may be written as a word in generators from $S^{(i)} = S\cap\Gamma^i$. In section 2.3 of their paper, [SS] supply a simple algorithmic loop for naming these cosets with words composed of generators in $S^{(i)}$. They call this loop, and the resulting factoring process, the Sims factoring algorithm.

For each Γ^i, the loop begins by finding its generating set $S^{(i)}=\{s_1^{(i)},\ldots,s_r^{(i)}\}$ by set intersection. Then each $s_j^{(i)}$ is applied to base point b_i to see where it is sent. Each image of these previous applications is in turn checked to see where each $s_j^{(i)}$ sends it, and so on until all possible results have been found as to where base point b_i may be sent by a word composed of elements from $S^{(i)}$. This method by exhaustion identifies all possible cosets (one for each value b_i may be sent to) of Γ^{i+1} in Γ^i, and names them with words from elements of $S^{(i)}$.

The above loop, then, develops a path from base point b_i to each of the possible values it may be sent to by a word in strong generators in $S^{(i)}$. Such a pattern may be represented by a tree T_i which has as its root the base point b_i and its remaining vertices the distinct values to which b_i is sent. The edge between a vertex x and its son vertex y will be labeled by the first strong generator found by the algorithm to have sent x to y.

Consider again the example of $A_4 \subset S_4$ with strong generating set $S=\{(12)(34),(123),(132)\}$ with respect to the ordered base $B=\{4,1\}$. Then for tree T_1, we consider all the values where base point $b_1=4$ may be sent by the generating set $S^{(1)}=S$, the entire generating set. Letting $a=(12)(34)$, $b=(123)$, and $b^{-1}=(132)$, and applying them in the order a, b, b^{-1} in each round, we see that $a(4)=3$, $b(4)=4$, and $b^{-1}(4)=4$. Reapplying to the only new value of 3 yields $a(3)=4$ (which is not new), $b(3)=1$, and $b^{-1}(3)=2$. Clearly, all possible values have been reached, so the algorithm would stop. For T_2, consider all the values to which base point $b_2=1$ may be sent by the generating set $S^{(2)}=S\cap\Gamma^2$. But Γ^2 is the subgroup of Γ where each element holds base point $b_1=4$ fixed. Thus $S\cap\Gamma^2 = \{b,b^{-1}\} = \{(123),(132)\}$. Applying b and b^{-1} in order to base point $b_2=1$ yields $b(1)=2$ and $b^{-1}(1)=3$. Clearly, 4 may not be reached, since these all hold 4 fixed, implying that if 4 were reachable by application of some generator, 4 could be moved by that generator's inverse. Hence, the algorithm stops since all possible values have been reached. (In practice, the algorithm would simply perform another loop only to find that no new values turned up.) Trees T_1 and T_2 are shown in Figure 5.2.4.

Each distinct path in tree T_i thus corresponds to the name (as a word in strong generators) of a distinct coset of Γ^{i+1} in Γ^i. Since all reachable values are represented as vertices, there is a one-to-one correspondence between these distinct paths and the elements of U^i. For example, in tree T_1 of Figure 5.2.4, the path from base point 4 through to vertex 1 corresponds to applying generator a followed by generator b. That is, any permutation in A_4 that sends 4 to 1 is in the coset named $b \circ a$.

Hence, it is clear from Theorem 5.2.3 and the above work that a factorization of any element of Γ may easily and efficiently be achieved for a Cayley graph based on a permutation group with a strong set of generators with

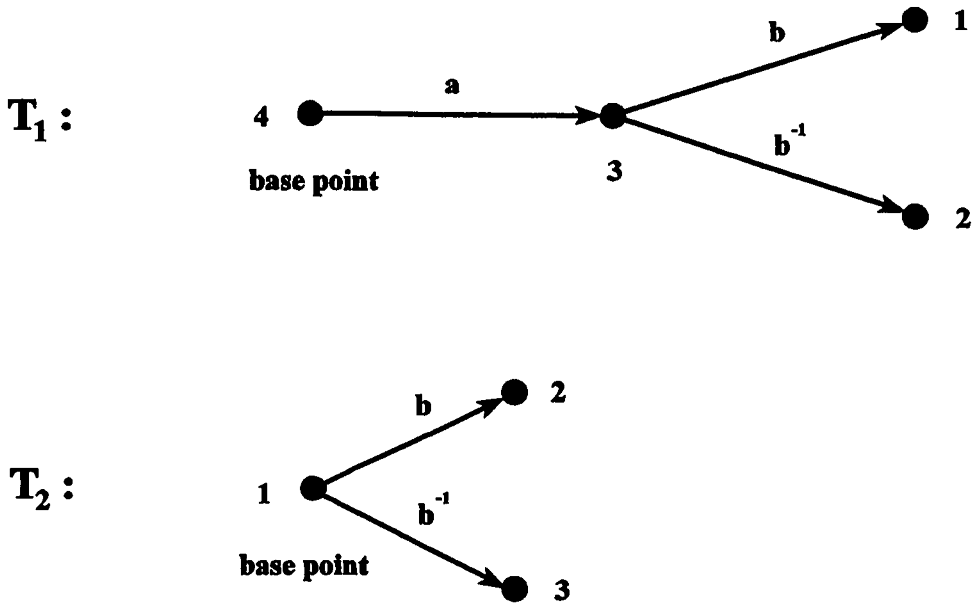

Figure 5.2.4: Trees produced by the Schibell and Stafford algorithm for the group $A_4 \subset S_4$ with strong generating set $S=\{(12)(34),(123),(132)\}$ with respect to the ordered base $B=\{4,1\}$.

respect to an ordered base.

5.3 The Routing Algorithm

Once such a Cayley graph has been selected as the model for the parallel-processor network architecture, each processor is assigned the name that is the permutation of its corresponding vertex in the Cayley graph. Each processor must store its own permutation name, the inverse of its name, and the full set of trees for the given ordered base. It is sufficient to note here that [SS] describe a method for computers to conveniently store the information in these trees in the form of Schreier vectors. For the purposes of this thesis, it is simpler to follow the algorithm using the trees.

As mentioned directly after the proof of Theorem 5.2.3, [SS] believe that the first edge named by their full algorithm will be the first edge needed to travel the path from vertex x to vertex y. In fact, this is not the case, and an adjustment to their algorithm must be made. To illustrate this problem, consider the following example from the previously shown group A_4 with the strong generating set $S=\{a=(12)(34), b=(123), b^{-1}=(132)\}$ with respect to the ordered base $B=\{4,1\}$, as used to find trees T_1 and T_2 in Figure 5.3.4. To find a path from x, the identity permutation, to the permutation $y=(134)$, clearly we have that yx^{-1} is the permutation y itself.

Hence, the task is to factor y using the tools just developed. By Theorem 5.2.3, we should be able to factor into $y=U_1 \circ U_2$, where U_1 is a word in strong generators which names the coset containing all the permutations sending 4 to the same place y sends it, and U_2 is a word in strong generators which names the coset of $(A_4)^2$ in A_4 containing all the permutations in $(A_4)^2$ sending 1 to where $[U_1]^{-1} \circ y$ sends it. Evaluating for the first base point, we see that $y(4)=1$, so we examine tree T_1 in Figure 5.3.4 and observe that application of b to 3 yields 1, and application of a to 4 yields 3, so that $b \circ a(4)=1$. (A simple calculation verifies that this is the case.) Then $U_1=b \circ a$, and we conclude that $U_2=[U_1]^{-1} \circ y$. But a calculation shows that

$$([U_1]^{-1}\circ y)(1) = ([b\circ a]^{-1}\circ y)(1) = (a^{-1}\circ b^{-1}\circ y)(1) = 2$$

so that U_2 is the path in T_2 which goes from 1 to itself. Hence, U_2 is the identity permutation, and the factorization is complete with $y=b\circ a$. A quick check verifies that this is indeed the case.

This factorization, of course, describes a path from x to y that is x--$a\circ x$--$b\circ a\circ x=y$. Certainly this path differs from one which would use edge a (the first strong generator found by the algorithm) then edge b (the next generator found). Were this latter path to be correct, it would certainly make routing in practice as simple as in theory:

> Simply take the first edge found, move along it to x', calculate where $y\circ[x']^{-1}$ sends the first base point, and repeat the process. If $y\circ[x']^{-1}$ doesn't move the current base point, check the next base point for movement, and use its Schreier vector (tree) to find the next edge. If no base point is moved, then $y\circ[x']^{-1}$ must be the identity permutation, and the trip is finished.

This process outlined by [SS] requires only that each processor check to see where the composition of y and its inverse send the current base point, and then check in its appropriate Schreier vector to find the next edge to travel. Unfortunately, it doesn't work this way.

Fortunately, however, there is a minor variation that allows this process to work much as [SS] envision. Suppose we found the complete factorization of $x \circ y^{-1}$ instead. From Theorem 5.2.3, we would obtain something of the form $U_1 \circ U_2 \circ \ldots \circ U_t$. Then this corresponds to a path from y to x along the sequence of edges U_t^*--...--U_2^*--U_1^*, where each section U_i^* corresponds to the reverse of the sequence of strong generators making up U_i. That is, if $U_i = s_{i1} \circ \ldots \circ s_{ik}$, then U_i^* is the sequence of edges s_{ik}--...--s_{i2}--s_{i1}.

Clearly, then, tracing this entire sequence in reverse is a path from x to y. If U_1 is a word in strong generators of length k, then the first k edges for this path from x to y are traced by going the "wrong way" along the sequence of edges s_{11}--...--s_{1k}: that is, by the path s_{11}^{-1}--...--s_{1k}^{-1}. However, s_{11} is the first strong generator found in tree T_1

when looking up the results of where $x \circ y^{-1}$ sent the first base point. Hence, a move along edge s_{11}^{-1} would be a first step along a path from x to y, and could be taken immediately after the first operation in calculating the path from y to x.

To see this, consider the following example. Let $H_1=(246)(357)$, $H_2=(167)(254)$. Then we have $H_1^{-1}=(264)(375)$ and $H_2^{-1}=(176)(245)$. Let F_{21} letters of order 21, generated by the strong generating set $S=\{H_1,H_2,H_1^{-1},H_2^{-1}\}$ with respect to the ordered base $B=\{1,3\}$. Suppose we wish to find a path from $x=(1576342)$ to $y=(154)(236)$. ($x=H_2H_1H_2$ while $y=H_1^2H_2^2H_1$, so both are in F_{21}.) A calculation shows $y \circ x^{-1}=(132)(467)$, and it sends base point 1 to 3. From tree T_1 in Figure 5.3.1, we get $U_1=H_1 \circ H_2^{-1}$. Then we find U_2 from tree T_2 of Figure 5.3.1 by determining that $U_1^{-1} \circ y \circ x^{-1} = (H_1 \circ H_2^{-1})^{-1} \circ y \circ x^{-1} = H_2 \circ H_1^{-1} \circ y \circ x^{-1}$ sends the second base point (3) to 7. Thus, $U_2=H_1^{-1}$. Then $y \circ x^{-1} = H_1H_2^{-1}H_1^{-1}$ (verified by calculation), and a path from x to y follows the sequence of edges H_1^{-1} to H_2^{-1} to H_1. Unfortunately, the entire sequence must be calculated initially before any moves can be made.

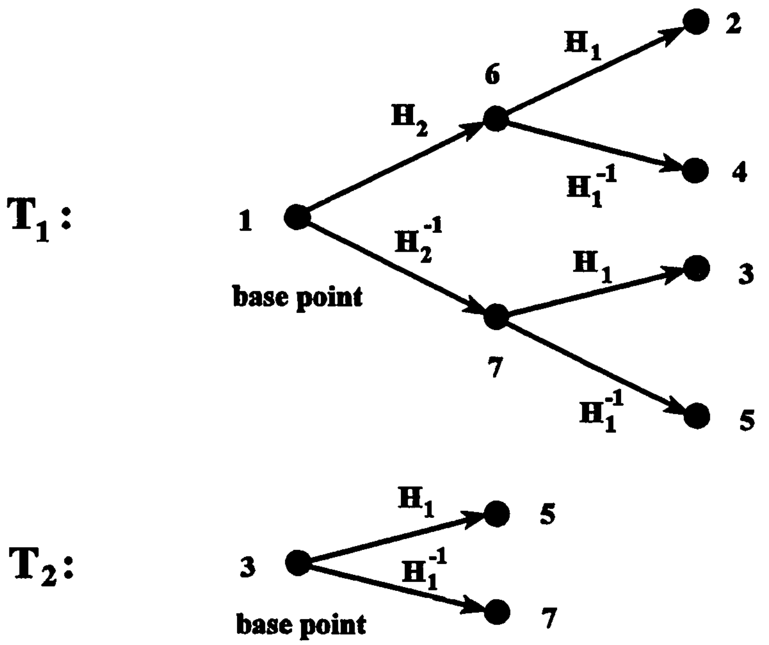

Figure 5.3.1: Trees produced by the Schibell and Stafford algorithm for F_{21}, the permutation group of order 21 with strong generating set $S=\{H_1, H_2, H_1^{-1}, H_2^{-1}\}$ with respect to the ordered base $B=\{1,3\}$. $H_1=(246)(357)$, $H_2=(167)(254)$, $H_1^{-1}=(264)(375)$, and $H_2^{-1}=(176)(245)$.

Contrast this result with the following "reverse" method. Calculate that $x \circ y^{-1}$ sends the first base point (1) to 2. From tree T_1, 2 is reached via H_1, so H_1^{-1} is the first step on the path from x to y. Determine that $H_1^{-1} \circ x \circ y^{-1}$ sends 1 to 6, which is reached (direct from 1) via H_2 in tree T_1. Thus, H_2^{-1} is the second edge to traverse in the path from x to y. Since 6 was reached direct from 1 in tree T_1, we are done finding U_1, so it is time to evaluate where $H_2^{-1}H_1^{-1}xy^{-1}$ sends the second base point (3). A calculation shows that it sends 3 to 7, which tree T_2 shows is reached (direct from 3) via H_1^{-1}. Then the last step in the path from x to y travels along edge $(H_1^{-1})^{-1}=H_1$. This path coincides perfectly with that found by the conventional method in the previous paragraph. Even were the paths different, the results would be the same: a path from x to y is found.

Clearly, this general-purpose routing algorithm is feasible for Cayley graphs based on permutation groups. The questions remain as to how efficient it is with regard to diameter, bottlenecking, and processor requirements.

5.4 Algorithmic Diameters and Other Concerns

Since the algorithmic loop in [SS] for finding the complete coset names works on a "first-reached" examination by exhaustion, the paths in the corresponding trees will be of minimum length for that particular strong generating set and ordered base. This suggests that the algorithmic diameters produced will be of reasonable size when compared to the actual true diameters.

The method of construction for the factorization of any element makes it clear that the algorithmic diameter of a particular Cayley graph is the longest word length (in strong generators) of any factorization of the set of elements in the underlying group. The longest such word will be the sum of the longest path lengths in each of the trees required. Since there is one tree for each base point, so, if we let $b=|B|$ and P_i be the length of the longest path in tree T_i, the algorithmic diameter will simply be

$$\text{AlgDiam}(G) = \sum_{i=1}^{b} P_i.$$

This suggests that an upper bound may be found in general, based on the fact that a tree with k vertices may not have a path length greater than k-1. Then, assigning n_i the value of the number of vertices in tree T_i, there is a strict upper bound on the algorithmic diameter of

$$\mathrm{AlgDiam}(G) = \sum_{i=1}^{b} (n_i - 1).$$

Since tree T_i cannot show a path to the first i-1 fixed base points, it can have at most k-(i-1) vertices, where k is the number of letters on which the permutations act. Hence, without any knowledge of the particular trees involved, if the cardinality of the ordered base is known, then

$$\begin{aligned}\mathrm{AlgDiam}(G) &= \sum_{i=1}^{b} [k-(i-1)] \\ &= b(k+1) - \sum_{i=1}^{b} i \\ &= b(k+1) - (b+1)b/2 \\ &= b(2k-b+1)/2. \qquad (*)\end{aligned}$$

In practice, the average tree produced by the Sims algorithm will not contain a path near as long as k-i+1, so this upper bound will not be very good. Note also that calculating average algorithmic diameter is a much more involved process. Even though it is possible to assert that the average path length in each tree cannot be more than half the worst case length of k-i+1, it may be the case that the longer paths are used more often in the factorization process for a particular graph, so this cannot be used to obtain an upper bound for the average algorithmic diameter.

To establish some feel for this algorithm, we consider the graphs used previously for examples. For A_4 with generating set and base as before, inspection of the Cayley graph yields a true diameter of 3. Summing the tree lengths from Figure 5.2.4, we also get an algorithmic diameter of 3. (The upper bound from equation (*) above is 7.) For our example of F_{21}, inspection of the Cayley graph shows a diameter of 3, and summing the tree lengths in Figure 5.3.1 yields an algorithmic diameter of 3. (The upper bound from equation (*) above is 13.) Certainly, for relatively small groups and good choices of generators and ordered bases, the algorithm can produce a very good (in these two cases, perfect) diameter.

However, [SS] provide an example of a larger group which demonstrates that the algorithm is not always so efficient. According to [SS], the sporadic simple Mathieu group M_{11} is of order 7920 and has a permutation representation of degree 12. They list a strong generating set of 8 elements with respect to the ordered base B={1,2,3,4}. Though they state this graph has diameter 7 and average diameter 5.25, they claim an algorithmic diameter of 12 and an average algorithmic diameter of 7.2. These values are respectively 71% and 37% above their corresponding true diameters. Unfortunately, as noted in chapter 6, no such algorithmic values are given for their proposed network architectures SS1, SS2, and SS3. As such, it is not possible to ascertain the efficiency of the routing algorithm in their cases, though the example of M_{11} suggests that there will be a sizable increase over the true diameters.

Regarding the problem of bottlenecks in the routing of information between processors, [SS] suggest that two methods of coping are possible. The first is based on the fact that it is usually possible to construct more than one set of coset representatives for each base point. As such, different trees would be produced for each base point. If more than one Schreier vector (computer format for the tree) were stored by each processor for each base point, then the distinct routing provided by each tree could be used to choose an alternate path in the case of a routing conflict. Though this would require more storage and an occasional extra look-up by the processor, it is a conceivable plan.

Alternatively, were the group to be constructed with a set of generators that are strong with respect to more than one base, alternative routings may be found by selecting the route provided by a different base. However, this extra constraint on group construction would often force the generating set to be larger than otherwise necessary, so an undesirable by-product of this adaptation is that the degree of this more-flexible graph would likely be higher.

Evaluating the effects on computer performance of all these possible extra storage and processing requirements is beyond the scope of this thesis. However, it is clear that some decrease in performance would result when compared to the simplicity of the hypercube's routing. [SS] reference a paper by Pittelli and Smitley [PS] that used some of their candidate architectures in a computer optimization study. Though the specific underlying assumptions of this study have not been examined, it seems that proposals of [SS] perform better than some of the current architectures in existence or under consideration at the time of the study. However, the hypercube was not included in this study due to a degree 6 constraint related to technology limits for maximum bandwidth of switches.

Overall, then, [SS] have proposed several interesting candidate architectures and an applicable general-purpose routing algorithm that could prove to yield a viable, efficient parallel-processing supercomputer. In Chapter 6, we suggest an alternative class of Cayley graphs and routing algorithm that may be competitive as supercomputer models.

CHAPTER 6

CAYLEY GRAPHS OF Z_n AS NETWORK MODELS

6.1 Introduction

As discussed earlier, much of the literature on network models is focused on trying to construct graphs with high expanding constants. This focus is based on the general relation that a highly expanding graph will have a low "branching-back" occurrence, behaving locally like a tree everywhere and thus resulting in as low a diameter as possible. The search for such graphs led many of these researchers into the realm of various rather exotic graphs. The k-sum and k-difference graphs of Chung [Ch] and the Ramanujan-style graphs of Lubotzky, Phillips, and Sarnak [LPS] are examples of this. Schibell and Stafford [SS] state that abelian groups do not fit the description of graphs with low "branch-back" occurrence, and instead turn to simple groups and Sylow-2 subgroups of exotic graphs with generating sets chosen to consist mainly or entirely of generators which are their own inverses.

Schibell and Stafford [SS] are the only ones of the literature surveyed to provide the diameters of specific graphs. A tabulation of some of their efforts as compared to other proposed and currently used graphs is shown below in Table 6.1.1.

Clearly, researchers have had some success in producing graphs of low degree and low diameter when contrasted with the benchmark hypercube. However, there are several problems to consider. First, as pointed out in [SS], the sparse distribution of the orders of simple groups makes it unlikely that there will be many such groups of the desired size to consider as candidates. Although there are quite a number of various groups of, for example, order 1024, it is no easy task to construct a group with favorable expansion characteristics and a specific size. Even should one succeed in doing so, the construction technique may not transfer easily to a group of a different size should that be desired. In short, good graphs may be hard to come by, and are limited in their useful sizes by design constraints.

TABLE 6.1.1 ([SS]): COMPARISON OF VARIOUS PROPOSED NETWORK MODELS TO THE HYPERCUBE BENCHMARK

Graph Format	Vertices	Degree	Diameter	Avg Diam
Hypercube	1024	10	10	5.0
Toroid (32 by 32)	1024	4	32	16.0
Toroid (8 by 8 by 16)	1024	6	16	8.0
Butterfly (128 by 8)	1024	4	10	6.6
Super Toroid	1024	4	12	6.8
SS1 : PSL(2,13)	1092	4	9	6.2
SS2 : subgroup of M_{24}	1024	5	8	5.2
SS3 : subgroup of S_{16}	1024	6	7	4.5

Note that SS1 through SS3 are the proposals of [SS], though SS1 is of the structure proposed by [LPS]. The other groups are generally known current proposed or existing architectures.

Of course, the above problem is moot once such a group is found of a desired size. The work is done, and the group has been constructed. Unfortunately, many of the groups shown in Table 6.1.1 accept a poor (high) absolute or average diameter for the advantage of lower degree. Though there are cost and construction advantages to lower degree (and, according to [SS], possibly some performance improvements related to data path width), increasing either (or both) diameter(s) beyond the benchmark of the hypercube indicates that it will be slower than the hypercube due to the higher number of clock ticks required to send information around the network.

However, the biggest problem with many of the exotic graphs presented in Table 6.1.1 is that the task of routing information between processors is cumbersome and inefficient. Certainly, Schibell and Stafford [SS] recognized this issue: they spend much of their paper trying to construct general purpose routing algorithms for such networks. Their efforts in this regard are presented in Chapter 5, but it is still in question whether their proposal is efficient enough.

First, their routing algorithm requires some programming and processor enhancements which may not be trivial. This might lead to significant cost increases for processors and a slowdown for information transfer due to increased complexity of routing calculations.

Second, the performance of a network with a true (average) diameter substantially superior to that of the hypercube, may, in fact, perform in inferior fashion if the routing algorithm required produces an algorithmic (average) diameter equal to or larger than that of the hypercube. In defense of this concern, note that [SS], while demonstrating their routing algorithm, use a Cayley graph on the Mathieu Group M_{11}, which has order 7920. Their calculations show that, although the true and average diameters are 7 and 5.25, respectively, the algorithmic equivalents are 12 and 7.2. These are increases by a factor of 1.7 and 1.4, respectively. Were these ratios to remain the same for the groups proposed by [SS] in Table 6.1.1, it is clear that the "functional" (algorithmic) diameter and average diameter of each group would be inferior to those of the hypercube. (The hypercube has essentially "perfect" routing: algorithmic and true diameters are equal, as discussed in Chapter 1.) Unfortunately, algorithmic values are not presented by [SS] for their proposals, so doubt remains on the subject of superior performance by SS1 through SS3, even though [SS] present experimental modelling results by Pittelli and Smitley [PS] which seem to support their claims.

6.2 Following the Simpler Path

With the above issues in mind, this thesis took the approach of trying to use clever choices of generators for more common groups in an effort to improve on the hypercube benchmark. Ultimately, attention focused on graphs with the underlying groups of Z_n configuration. Clearly the issue of constructing groups of a desired size is no longer a concern: groups of any size may be chosen. In addition, the generating set may be chosen to be of any even size if n is odd, or of any size whatsoever if n is even (using n/2 as a generator to get a generating set of odd size). Hence, for any desired n, there exists extensive flexibility to trade off the degree with the diameter. For example, choosing to match the degree of the equivalently-sized hypercube makes direct comparisons of the diameters a valid process. Taking the opposite tack of constructing a graph with the same diameter as the hypercube allows direct comparison of their degrees.

However, the major strength of this approach really becomes apparent when the simplicity of the routing algorithm is taken into consideration: since for virtually all cases the routing method is the well-known, extremely simple "greedy" algorithm, the algorithmic average and full diameters are found to be either identical or very close to the true diameters. Hence, elaborate processors are not required, and the routing algorithm is "perfect" in the same sense as that of the hypercube's, thus validating direct comparison of diameters. In essence, the decision to pursue this approach depends on the assumption that, in order to improve on the hypercube, there is no need to create graphs with wonderful true diameters if, instead, it is possible to create graphs with merely good diameters yet perfect routing algorithms. At least when compared to the hypercube, this assumption proves to be valid.

Finally, in what appears to be a concern for current research in parallel-processing computers, increasing or decreasing the number of processors in an existing machine would not be a difficult task, although the resulting machine may not be as well-optimized as the original construction for the new number of nodes. Certainly, it would be much simpler to perform than on the hypercube, and would allow full flexibility in terms of the number of nodes added or subtracted, something the hypercube construction does not allow. (In the hypercube, the number of processors may only go up or down by a power of 2 in order to maintain its structure.) Basically, the physical structure of a network with Z_n-based Cayley graphs is quite simple to construct and adapt.

6.3 Basic Routing in a 4-Regular Cayley Graphs on Z_n

This section shows the evolution of how to best represent, and route in, the basic 4-regular Cayley graph, since keeping the generating set small allows for simpler analysis of the graph's properties. First examined were Cayley graphs based on groups of the form Z_{pq}, where p and q are distinct primes. Consider the graph of Z_{21}, where p=3 and q=7. Using the generating set $S = \{3,7,-3=3^{-1},-7=7^{-1}\} = \{\pm3,\pm7\}$, this Cayley graph may be represented by a sequence of 21 vertices (labeled from 0 through 20) equally spaced around a circle, with each vertex connected by an edge to the vertices both ±3 and ±7 places away from its position in the circle. Figure 6.3.1 shows this Cayley graph representation.

Unfortunately, it is clear from examining this graph that any routing algorithm is a mess. By exhaustion of possibilities, the diameter is 4, but there is no smooth way of predicting the shortest route to any particular vertex from any other. This issue led to reconceptualizing the way of representing Z_{21} with those generators: consider Z_{21} as $Z_7 \times Z_3$, label vertices as ordered pairs, and set 7=(1,0) and 3=(0,1). This representation is shown in Figure 6.3.2.

Clearly, routing on this graph is much simpler: from any vertex, one travels the shortest route in Z_7 by $\pm(1,0)$ to the value of the first entry of the target vertex's ordered pair, then moves in Z_3 to match the second entry. Such a routing method may be viewed as nearly identical to that of the hypercube: move by ones to match the entries as written in some m-tuple.

One problem with such a Cayley graph, however, is that it has one of the same major limitations found with the hypercube. For example Z_{21} may only be viewed as $Z_7 \times Z_3$ or as Z_{21} itself. Hence, other generators cannot be considered as candidates in order to ascertain whether advantage may be taken of certain structural characteristics. Z_{17} would be even more severely limited in choices.

A third alternative for a Cayley graph of Z_{21} combines some aspects of both the previous two, yet opens the door wide for flexibility in generating sets and routing. Initially, the routing in Figure 6.3.2 strikes one as taking jumps around the graph to get to the proper "cluster" of three vertices, and then finishing the route by moving to the desired vertex of that cluster's triangle. However, it may also be viewed as moving to the starting triangle's vertex contained in the desired 7-gon, then tracing the shortest route around the 7-gon to the destination vertex.

If Z_{21} is redrawn in circle form as in Figure 6.3.1, this routing is similar to using a generating set of $S=\{\pm1,\pm7\}$ to get to the proper multiple of (±)7, then finishing the trip with (±)1's to reach the final vertex. Algorithms similar to this routing method are often known as "greedy" algorithms, because they take the greediest bites possible (here, jumps of 7) to zero in quickly on the target area, then finish up with "dainty" bites (jumps of size 1). Such a graph is shown in Figure 6.3.3.

On the surface, there is no improvement here over the routing for Figure 6.3.2. However, Figure 6.3.2 depends on the size of the group being a multiple of the size of the n-gons. Hence, its only possible shape is nested 7-gons and triangles. This constraint does not apply to Figure 6.3.3. The generating sets $S=\{\pm1,\pm6,\}$ or $S=\{\pm1,\pm8\}$ may just as easily be used. In addition, a group of **any** size may be

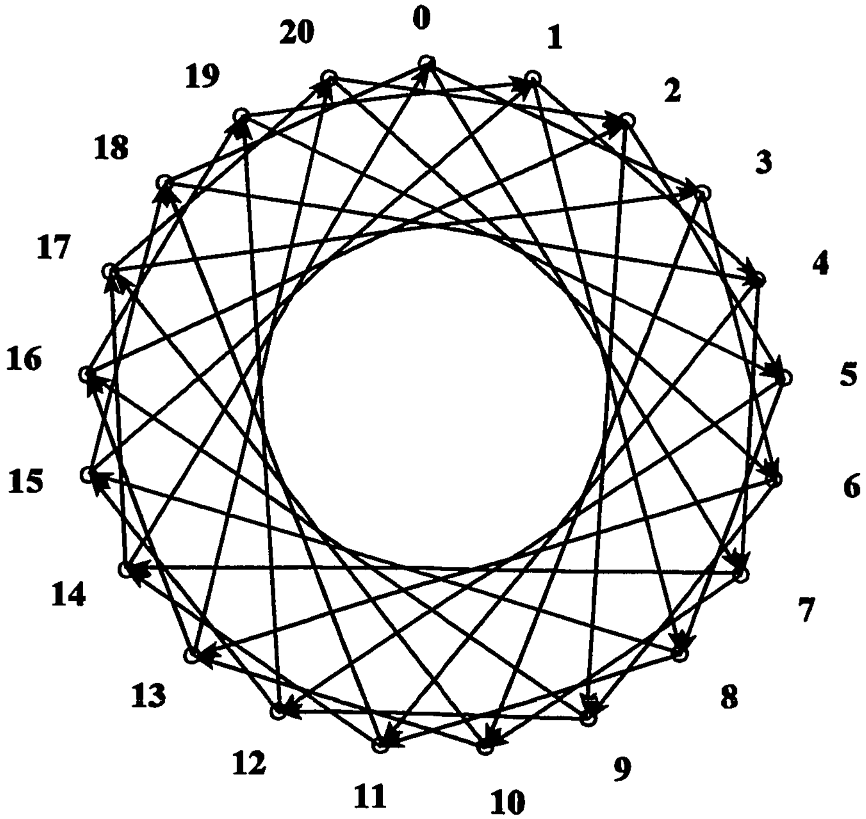

Figure 6.3.1: The 4-regular Cayley graph on Z_{21} with generating set $S=\{\pm 3, \pm 7\}$.

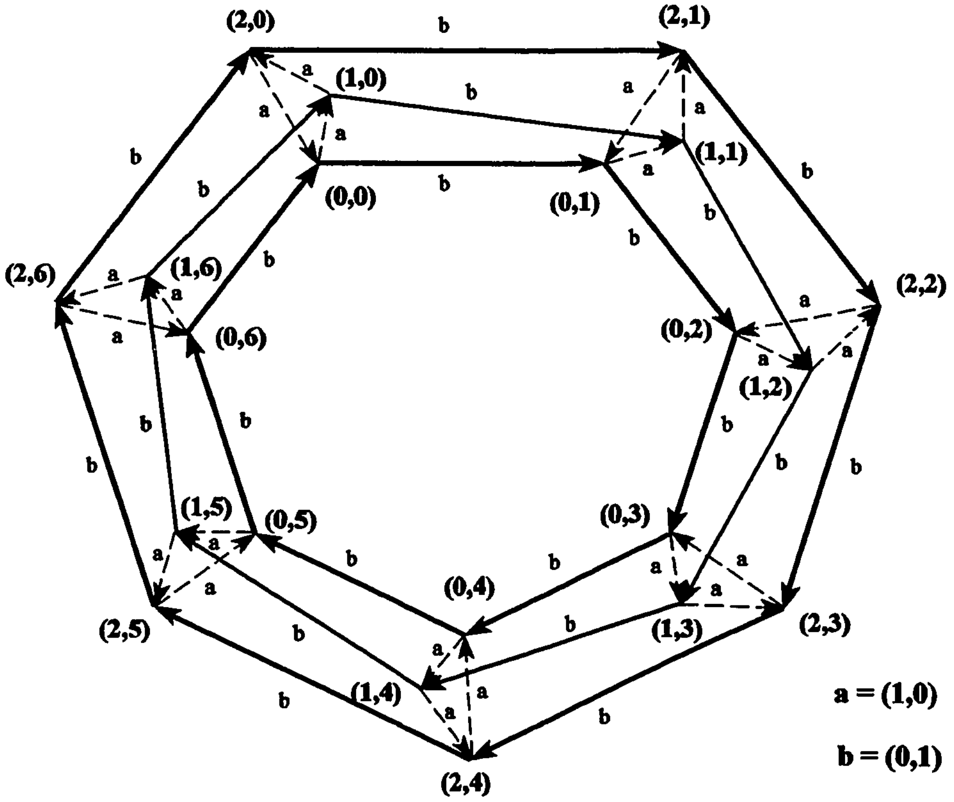

Figure 6.3.2: The 4-regular Cayley Graph of Z_{21} with $S=\{\pm3,\pm7\}$ as generating set. The labeling here views Z_{21} as isomorphic to $Z_3 \times Z_7$ with generators a=(1,0) and b=(0,1) and their inverses.

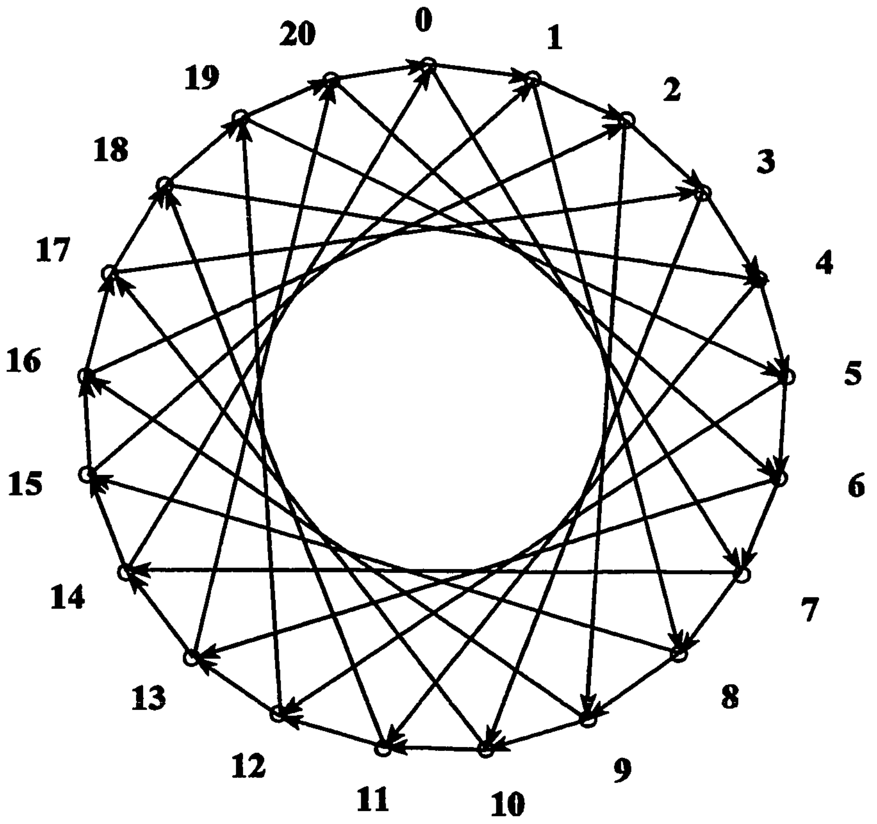

Figure 6.3.3: The 4-regular Cayley graph on Z_{21} with generating set $S=\{\pm1,\pm7\}$.

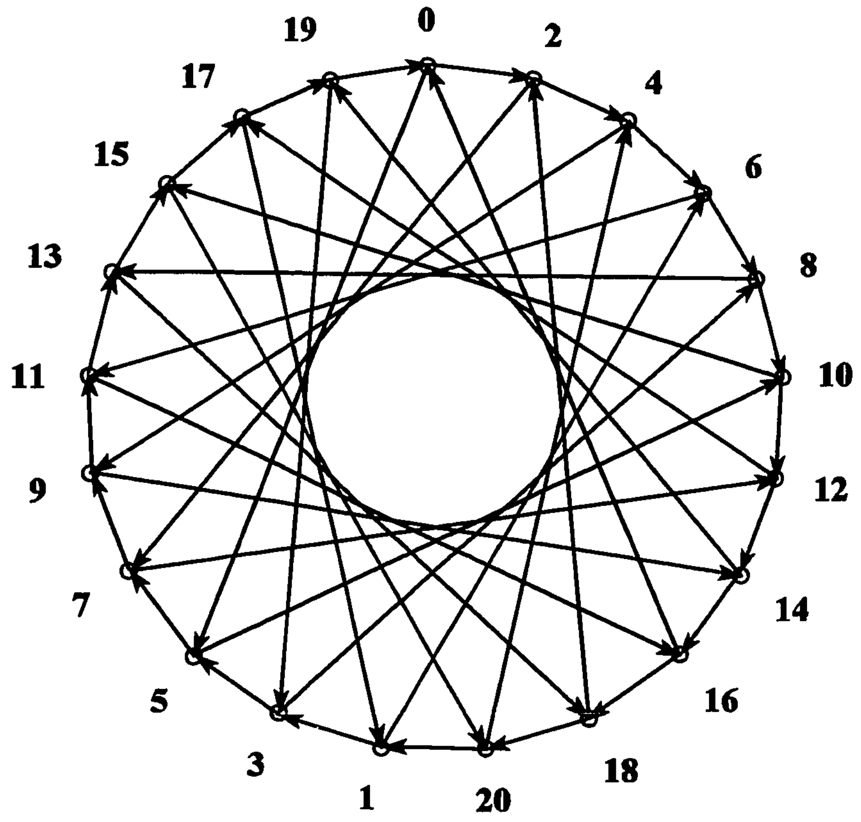

Figure 6.3.4a: The 4-regular Cayley graph on Z_{21} with generating set $S=\{\pm2,\pm5\}$, redrawn with vertices connected by ±2 shown as adjacent in the circle.

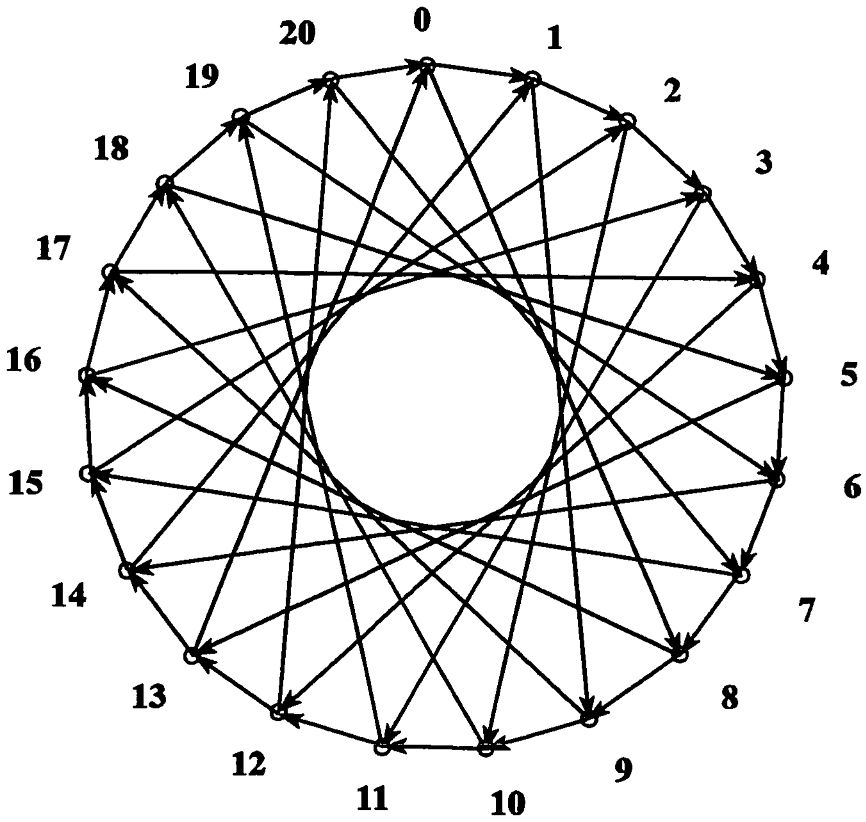

Figure 6.3.4b: The 4-regular Cayley graph on Z_{21} with generating set $S=\{\pm1,\pm8\}$. Note that it has the identical shape to that of the Cayley graph with generating set $S=\{\pm2,\pm5\}$ shown in Figure 6.3.4a.

used, regardless of its factor set, with total flexibility in choice of generating sets. The only requirement is that ±1 is included in the generating set. This ensures that S does indeed generate Z_n and allows the use of a straightforward greedy algorithm.

It may at first appear that this is still too restrictive on S. After all, Z_{21} may be generated by the set $S=\{\pm3,\pm7\}$. However, this generating set does not allow the employment of a simple routing algorithm, while any set containing 1 will be suitable for the greedy algorithm. Note also that forcing 1 to be in the generating set is merely equivalent to saying that one of the elements in S must generate Z_n by itself, a seemingly much less restrictive requirement. To see this, consider Z_{21} with generating set $S=\{\pm2,\pm5\}$. Since 2 will generate Z_{21} by itself, this Cayley graph may be redrawn as a circle of 21 vertices labeled in sequence counting by 2's (since such vertices would be adjacent to each other), with every pair of vertices whose labels differ by 5 connected by an edge. Figure 6.3.4a would be the result.

However, in Figure 6.3.4a, vertices in sequence are adjacent and the long jumps span a gap of 8 vertices when counting around the circle. Hence, Figure 6.3.4a has exactly the same shape as Z_{21} with generating set $S=\{\pm1,\pm8\}$ drawn in the conventional manner shown in Figure 6.3.4b.

Hence, for the 4-regular case, such a construction method appears a good candidate to allow any size of underlying group desired, efficient routing, and a relatively flexible choice of generating sets of the form $S=\{\pm1, \pm y\}$.

6.4 Constants for Cayley Graphs on Z_n with $S=\{\pm1,\pm\lfloor n^{1/2}\rfloor\}$

A logical choice enabling y to be a fairly ideal "partner" for 1 in the generating set seemed to be a value close to $n^{1/2}$. Consider if n=100 and y is chosen too large, say $S=\{\pm1,\pm30\}$, then the greedy algorithm makes big jumps with 30, but it will take up to fifteen steps to close on a target vertex from there. Conversely, if y is too small, say for $S=\{\pm1,\pm5\}$, then it takes too many jumps of size 5 to get half-way around the circle. This reasoning, plus the influence of the fact that f_k (used in Chapter 3 to determine the Kazhdan constant) always seemed optimal at $\{1,\lfloor n^{1/2}\rfloor\}$ for the 4-regular case, prompted the initial study of $S=\{\pm1,\pm x\}$, where $x=\lfloor n^{1/2}\rfloor$, the greatest integer not exceeding $n^{1/2}$. In other words, x is the truncated square root of n. This led to the following theorem regarding the diameter of such Cayley graphs.

THEOREM 6.4.1: Let G be the Cayley graph on Z_n, with generating set $S=\{\pm1,\pm x\}$, where x is the truncated square root of n. Then

$$\mathrm{Diam}(G) \leq x.$$

Proof: View the Cayley graph as a circle of n vertices, with each vertex adjacent to its neighbors in the circle and also adjacent to each vertex x vertices away in the circle. Since Cayley graphs are symmetric and therefore appear the same from every vertex (see Chapter 1), finding the maximum distance from vertex 0 to any other vertex establishes the diameter. Consider the following two cases.

Case 1: x is odd.

Then both the clockwise and counter-clockwise paths tracing $(x+1)/2$ steps from vertex 0 along the edges which jump x vertices at a time will have landed on or jumped past

$$x[(x+1)/2] + x[(x+1)/2] = x^2+x$$

total vertices. Counting vertex 0, this trip will thus have "spanned" a total of x^2+x+1 vertices. ("Spanned" is used here to mean that a vertex was either landed upon or jumped over during this trip.) Since x is the truncated square root of n, we have that

$$x^2 \leq n \leq (x+1)^2 - 1 = x^2+2x.$$

Observing that

$$n - (x^2+x+1) \le x^2 + 2x - (x^2 + x + 1) = x-1$$

we have that there are at most x-1 unspanned vertices remaining. Thus, any of these vertices is at most (x-1)/2 from the end vertex of either the clockwise or counter-clockwise path. In a similar argument, there are x-1 spanned vertices between each vertex that was landed on (a "landing pad") in our two paths. Clearly, then, any vertex between landings is at most (x-1)/2 short edges away from one of the landing pads. The result of all this is that any vertex in the Cayley graph is at most

$$(x+1)/2 + (x-1)/2 = x$$

steps away from vertex 0.

Case 2: x is even.

By a similar argument, tracing either a clockwise path or a counter-clockwise path from vertex 0, each of length x/2 + 1, along edges which jump x vertices at a time will span all the vertices of the Cayley graph, since

$$x(x/2 + 1) + x(x/2 + 1) = x^2+2x = (x+1)^2 - 1 \geq n.$$

Since there will be x-1 vertices between each pair of landing pads, we can see that the midpoint of those x-1 vertices will be x/2 steps from either of the landing pad vertices. Since either landing pad vertex will do, choosing the first one landed upon means that any "midpoint" vertex may be reached within

$$x/2 + x/2 = x$$

steps. Clearly, any "between" vertex that is not a midpoint vertex is within x/2 - 1 short edges of a landing pad vertex, so we have as the worst case that such a vertex will be a distance

$$(x/2 + 1) + (x/2 - 1) = x$$

away from vertex 0.

Hence, the diameter is less than or equal to x for such Cayley graphs. □

This result begs the obvious question: is x the lower bound for the diameter as well? In addition, is the diameter better with a generating partner for 1 other than x? Unfortunately, the answer to both these questions is: "It depends." After examining a number of Cayley graphs for values of n ranging from 10 to 50, it is clear that for some of these values, the diameter of the Cayley graph using $S=\{\pm 1,\pm x\}$ was indeed equal to x and no partner other than x could improve upon this diameter.

However, many graphs do not follow this pattern. For example, Z_{21} has a diameter of 3 when $S=\{\pm 1,\pm 8\}$, yet x=4. Z_{23} has a diameter of 3 when $S=\{\pm 1,\pm 5\}$, yet x=4. In addition, Z_{25} with $S=\{\pm 1,\pm 5\}$ has a diameter of 4 which is less than x=5, even though ±5 is the generating partner. Similarly, for Z_{10} a choice of $S=\{\pm 1,\pm 3\}$ yields a diameter of 3 while the choice of $S=\{\pm 1,\pm 4\}$ yields a diameter of 2. It is also interesting to note that S={±1,±4} does not produce as large a Kazhdan constant as S={±1,±3} (f_k=2 compared with f_k=3) yet produces a Cayley graph with a smaller diameter. There are a number of other examples demonstrating that optimal Kazhdan numbers and optimal diameters do not necessarily correlate.

Table 6.4.2 shows the results of the exploration of the diameters achieved for various generating sets of size 4 for the values of n={10,15-25,32,35,36,45,48,49}. It should be clear from the results that there is no simple answer for any given n as to whether or not x can be improved upon for the diameter. For n=20, 24, 35, 36, and 48, no generating sets were found which provided a diameter less than x. For each of the remaining values, at least one generating set was found which yielded a diameter of x-1. The choice of generating sets to reach these improved diameters ranged from {±1,±x} itself, to {±1,±(x+1)} to something entirely different, though it seems worth noting that {±1,±(x+1)} worked most of the time. Also, as the value of n approaches $(x+1)^2$, it should be more difficult for the diameter to be less than x. This proved to be the case for Z_{24}, Z_{35}, and Z_{48}.

TABLE 6.4.2: BEST ACHIEVABLE DIAMETERS AND λ_1 FOR 4-REGULAR CAYLEY GRAPHS ON Z_n WITH S={±1,±y}

n=	10	15	16	17	18	19	20	21	22
Best Diameter	2	3	3	3	3	3	3	3	3
Examples of gen. sets for best diameter	(1,4)	(1,3) (1,4) (1,5)	(1,4) (1,6)	(1,4) (1,5)	(1,4) (1,5)	(1,4) (1,5)	(1,8)	(1,8)	(1,6)
Best λ_1	3.00	2.38	2.00	1.95	2.00	1.62	1.77	1.51	1.49
All generating sets for best λ_1	(1,3)	(1,4)	(1,4)	(1,4)	(1,5) (1,7)	(1,4) (1,5)	(1,8)	(1,6) (1,8)	(1,5) (1,6) (1,9)
λ_1 if S=(1,x)	3.00	1.56	2.00	1.95	1.77	1.62	1.48	1.36	1.25

(TABLE 6.4.2 cont'd)

n=	23	24	25	32	35	36	45	48	49
Best Diameter	3	4	4	4	5	5	5	6	6
Examples of gen. sets for best diameter	(1,5)	(1,4) (1,5) (1,6)	(1,5) (1,6) (1,7)	(1,7)	(1,6) (1,5)	(1,6)	(1,6) (1,7) 1,19	(1,6) (1,7) (1,8)	(1,6) (1,7) (1,8)
Best λ_1	1.67	1.55	1.47	1.17	1.08	1.00	.94	0.80	0.75
All generating sets for best λ_1	(1,5) (1,9)	(1,5)	(1,7)	(1,7) (1,9)	(1,6)	(1,6)	1,19	(1,7)	(1,7) 1,14 1,21
λ_1 if S=(1,x)	1.15	1.07	1.38	0.93	0.79	1.00	0.68	0.60	0.75

In addition, for those values of n which approach $(x+1)^2$, it seems that a strictly "greedy" algorithm likely will not achieve the true diameter; that is, often one must include a path to a vertex which uses the larger entry to come in from the "back side" of the graph's circle. For example, in Z_{21}, in order to yield a diameter of 3 with set $S=\{\pm1,\pm8\}$, the vertex 4 may be landed on only by a route of 0-13-5-4, this being a sequence of adding -8 twice, then -1.

Similarly, such a "backdoor" route is the only way that Z_{22} using $S=\{\pm1,\pm6\}$ can have a diameter of 3 instead of x=4. Hence, even though the true diameter is better, the routing algorithm will be more complex to achieve such a value. With the conventional "greedy" algorithm, both of these graphs will have algorithmic diameters of 4.

It is of interest to note from Table 6.4.2 that for each tested value of n except n=10, at least one of the generating sets producing the optimal (highest) λ_1 also produced a Cayley graph with the best achievable diameter found. This could be a useful correlation for finding true diameters of larger groups with larger generating sets if it holds for such groups, and if such a correlation can be easily adapted into a computer program to search all possible generating sets.

It seems likely that a generating set of the form $S=\{\pm1,\pm y\}$ will yield the best λ_1 for each value of n, though this has not been proven. The alternatives for such sets must be of the form $S=\{\pm p,\pm q\}$, where gcd(p,q) generates Z_n yet gcd(p,n)>1 and gcd(q,n)>1. There is a fairly limited choice of such sets for each $n\leq50$, based on its factor sets, so a search by exhaustion could be an interesting test of this idea. For example, $S=\{\pm3,\pm7\}$, $S=\{\pm6,\pm7\}$, and $S=\{\pm9,\pm7\}$ are the only such candidates for Z_{21}. A search of these generating sets showed each Cayley graph has the same value of $\lambda_1=.753$, which is substantially inferior to the value of λ_1 in Table 6.4.2.

However, to reiterate, if the true diameter requires the use of an algorithm that is an exhaustive search of all possible paths, including the "backdoor" routes discussed above, it is not a useful result. That is, an improvement of diameter by only 1 is not worth a major blow to the efficiency of the routing. If a different choice of generating sets yields an improvement of the diameter by 1 (or more) using the "greedy" algorithm, then it is a better result; otherwise not.

THEOREM 6.4.3: Let G be the Cayley graph on Z_n, with generating set $S=\{\pm 1, \pm x\}$, where x is the truncated square root of n. Then the value of the Cheeger constant h is

$$h(G) = \begin{cases} 4x/n & \text{if n is even} \\ 4x/(n-1) & \text{if n is odd.} \end{cases}$$

Proof: The definition of the Cheeger constant h for a specific graph G=G(V,E) is

$$h(G) = \mathit{min}\{|E(A,B)|/(\mathit{min}\{|A|,|B|\}: A,B \text{ partition } V\},$$

where E(A,B) is the set of "bridge" edges with one end in A and the other in B. Then the value of h occurs with the worst case partitioning of G. (Note that for a Cayley graph to be 4-regular, it must have at least 5 vertices. Thus, $n \geq 5$ in all cases.)

Case 1: $1 \leq |A| < x$.

No matter how such a partition is made, each vertex in set A will be connected to at least 2 vertices in set B, thereby yielding a candidate h that is

$$h \geq 2|A|/|A| = 2.$$

Case 2: $x \leq |A| \leq n/2$.

Again, no matter how such a set A is chosen, there will always be at least 2x edges connected to the vertices in set $B = V-A$. In an attempt to make all the vertices in A adjacent along the edges of 1 and 1^{-1}, the nearest x vertices on each end of the string of |A| vertices will be adjacent along the edges x and x^{-1}. An attempt to make all the vertices of A adjacent through the edges of x and x^{-1} produces "stars" of x vertices; positioning k of these "stars" next to each other -- i.e., one vertex clockwise or counter-clockwise from each other -- will yield x "strings" of k vertices adjacent along the short edges, where each of these "strings" will have an adjacent vertex from set B on each end. Any mixture of these two methods will increase the number of bridge edges.

Since we must have at least 2x neighbors in this case, the worst case would be when $|A|=n/2$ (if n is even) or $|A|=(n-1)/2$ (if n is odd). Hence, in this case we have

$$h \geq 2x/(n/2) = 4x/n \quad \text{or} \quad h \geq 2x/[(n-1)/2] = 4x/(n-1).$$

Since $x \leq n^{1/2}$, and $n \geq 5$, we have that $x/n \leq 1/2$. Thus, Case 2 provides a lower bound for h and we have that

$$\begin{array}{lll} & h \geq 4x/n & \text{if n is even} \\ \text{or} & h \geq 4x/(n-1) & \text{if n is odd.} \end{array}$$

However, if we choose the set A such that it consists of a string of n/2 or (n-1)/2 adjacent vertices (along the short edges of 1 and 1^{-1}), then we get a value for h that is, in fact, equal to this lower bound. Hence:

$$h(G) = \begin{cases} 4x/n & \text{if n is even} \\ 4x/(n-1) & \text{if n is odd.} \end{cases}$$ □

THEOREM 6.4.4: Let G be the Cayley graph on Z_n, with generating set $S=\{\pm1,\pm x\}$, where x is the truncated square root of n. Then

$$\lambda_1 = \begin{cases} 2-2\cos(2\pi/n^{\frac{1}{2}}) & \text{if n is a perfect square;} \\ 4-2[\cos(2\pi/n)+\cos(2\pi x/n)] & \text{otherwise.} \end{cases}$$

***Proof*:** Given in Appendix D. □

Recalling from Theorem 4.3.4 that $c = 2\lambda_1/(k+2\lambda_1)$, we see that the constant c may be easily evaluated when $S=\{\pm1,\pm x\}$. Clearly these results are lower bounds for the expansion constant c for the graph using the best generating set possible. However, as may be seen in Table 6.4.2, the best value for λ_1 for 4-regular Cayley graphs on Z_n is not significantly better, and likely will tend toward zero as n gets large, just as the value for λ_1 tends toward zero when $S=\{\pm 1, \pm x\}$.

This brings us unswervingly toward the following conclusion: for the family of graphs on Z_n with $S=\{\pm 1, \pm x\}$ it is apparent that **every** significant constant we have established in the theoretical work tends toward zero as n gets very large, with the obvious exception of diameter, which gets very large as n increases. To recall, diameter is proportional to $n^{½}$, h is inversely proportional to $n^{½}$ (and thus both Kazhdan constants are inversely proportional to $n^{1/4}$), and λ_1 drops as $\cos(2\pi/n^{½})$ approaches 1.

On this note, it is also apparent that such a family will not be Ramanujan, for which the requirement was that $\mu \leq 2(k-1)^{½}$. Recalling that $\mu \geq k-\lambda_1$, we observe that a graph will certainly not be Ramanujan if $\lambda_1 < k-2(k-1)^{½} = 4-2(3)^{½} \approx 0.535$. Since as n gets very large, λ_1 tends toward zero, there is no hope of constructing a Ramanujan family out of the Z_n's for $S=\{\pm1,\pm x\}$. Indeed, the chances seem remote that allowing any generating set of a fixed size k will possibly provide an infinite family of Ramanujan graphs, given the trends for the case when $S=\{\pm1,\pm x\}$.

6.5 Cayley Graphs on Z_{32} vs. the Hypercube on $n=2^5$

This section compares the characteristics of Cayley graphs based on Z_{32} with the performance of the hypercube of the same size. We choose n=32, since this conforms to the hypercube size of 2^5, yet is still a manageable size for examination by exhaustion.

From Theorem 6.4.1, it is possible to construct a 4-regular Cayley graph on Z_{32} with diameter less than or equal to $\lfloor 32^{½} \rfloor=5$. Since the hypercube on n=32 has degree 5 and diameter 5, this would seem to be a promising start. Table 6.5.1 compares the diameter, average diameter, algorithmic diameter, and average algorithmic diameter of several such Cayley graphs to those of the hypercube.

TABLE 6.5.1: COMPARISON OF Z_n GRAPHS vs. HYPERCUBE, FOR n=32

Graph Design and Generating Set	Degree	Diam.	Avg. Diam.	Algor. Diam.	Avg. Alg. Diam.
Hypercube ($32=2^5$)	5	5	2.5	5	2.5
Z_{32} S={±1,±4}	4	5	2.88	5	2.88
Z_{32} S={±1,±5}	4	5	2.75	5	2.75
Z_{32} S={±1,±6}	4	5	2.66	5	2.78
Z_{32} S={±1,±7}	4	4	2.63	5	2.75
Z_{32} S={±1,±4, 16}	5	4	2.28	4	2.28

Each diameter and average diameter is the best that can be found, even allowing for "backdoor" routes. Note that for S={±1,±7}, the backdoor route provides a lower true diameter. Algorithmic values are using strictly greedy algorithms, with no backdoor routes allowed. Since 16 is its own inverse in Z_{32}, it contributes only one generator to the size of S.

It is encouraging to observe that, by using S={±1,±4,16}, we can match the degree of the hypercube for n=32 and improve upon its algorithmic diameter and average algorithmic diameter performance. Conversely, we can match the diameter with a graph of lower degree, though there is a small sacrifice in average algorithmic diameter performance. This suggests that if we can find a model for construction of larger graphs (higher n) using these as building blocks, we will have the flexibility in design to improve on the hypercube's performance in either way desired.

Finding the true diameter and average diameter of many graphs of the form Z_n with $S=\{\pm s_i\}$ is a difficult task due to the presence of "backdoor" routing opportunities to reduce path lengths between vertices. Since these opportunities tend to be quite case-specific, the only obvious sure-fire method will be examination by exhaustion. For the graphs in Table 6.5.1, this is easily done, and those values are shown in the columns labeled Diameter and Average Diameter. However, for larger values of n, these values will not be found.

Instead, we focus on the more important measures of the performance of the graph as a processor network: the algorithmic average and full diameters. It is a straightforward matter to calculate these algorithmic diameters, as the "greedy" routing algorithm forms easily recognized patterns of the number of steps required to reach all other vertices. Finding the highest value in the pattern of a particular graph clearly identifies the algorithmic diameter of that graph. To calculate the average algorithmic diameter of a graph, we need a weighted average for the pattern of the number of steps required to reach each target vertex starting from zero. This may be arrived at by the following formula:

$$\sum_{i=0}^{n-1} (N_i)(F_i),$$

where N_i is the number of steps required to reach vertex i and F_i is the fraction of time that vertex i would be the desired target vertex.

Since each processor (vertex) should theoretically be the target an equal fraction of the time, in simple cases this formula reduces to summing the number of steps and dividing by n. To find the number of steps to reach a given target vertex, we view the Cayley graph of Z_n with full generating set $S'=\{-s_t,\ldots,-s_2,-s_1=-1,s_1=1,s_2,\ldots,s_t\}$ as a circle of n vertices. Each vertex is connected by an edge to its immediate neighbor (since ±1 is in the generating set), its $\pm s_2^{th}$ neighbor, its $\pm s_3^{th}$ neighbor, and so on up through its $\pm s_t^{th}$ neighbor. Without loss of generality, we may assume that the starting vertex is at the top and is labeled zero. Going clockwise, label the next n/2 (or (n-1)/2, if n is odd) vertices in sequence from 1 through n/2 (or (n-1)/2, for n odd), while those in the counter-clockwise direction are labeled -1 through -[(n/2)-1] in sequence.

The basic "greedy" algorithm will search out a path to a target vertex by the following strategy:

1. By subtracting the conventional label of the starting vertex (call it b) from the label of the target vertex, and converting that value to modulo n in symmetric format, it is equivalent to assuming that the starting vertex is at zero and the target is some known distance away that is less than or equal to n/2 and the shortest direction (clockwise or counter-clockwise) is known. In short, we have the situation described in the paragraph above. For example, in Z_{32}, if you began at vertex 7, and the target vertex was labeled 20, this is equivalent to saying that the target is +13 vertices away, or 13 vertices in the clockwise direction.

However, if the target vertex were labeled 26, then it would be -13 vertices away, or 13 vertices in the counterclockwise direction.

2. Let c be this distance from the starting vertex to the target vertex. If c=0, the target vertex is the starting vertex. Otherwise, find $r=\pm s_i$, the largest (in absolute value) generator of the same sign as c such that $|c| > |r|/2$. This finds the optimum generator to travel on in order to move closer to the target vertex. Jump to vertex b+r, and return to step 1, replacing b with b+r, the label of the new starting vertex.

Below are the breakdowns for the diameters of the Cayley graphs in Table 6.5.1 based on Z_{32}. These graphs are small enough to list the number of steps required to land on each vertex. Again, since each individual vertex is equally likely as a target vertex, this category requires only that we sum the list of required steps and divide by the number of vertices. Table E.1 in Appendix E provides such lists for many of these smaller Cayley graphs. Note that only the positive values are listed; the negative values contribute equally, though we must be careful not to count 0 and n/2 twice, as we do for those in between. Such totals will play a role as building blocks in analyzing the averages for more complicated graphs.

Example 6.5.2: Z_{32} using $S=\{1,7\}$.

Table 6.5.3 shows an excerpt from Table E.1 specific to this Cayley graph. Landing on vertices ±1 through ±15 contribute $2 \cdot 42=84$ steps if the greedy algorithm is used or $2 \cdot 40=80$ if "backdoor" routes are allowed. Reaching vertex 16 takes 4 jumps in either case, while staying at vertex zero takes no jumps. Hence, we get an average true diameter of $84/32=2.625$ and an average (greedy) algorithmic diameter of $88/32=2.75$. Observing that the maximum number of steps

TABLE 6.5.3: NUMBER OF STEPS REQUIRED FROM 0 TO TARGET VERTEX IN Z_{32} USING S={±1,±7} AS GENERATING SET

Target	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Σ
Greedy	0	1	2	3	4	3	2	1	2	3	4	5	4	3	2	3	4	46
Backdoor	0	1	2	3	4	3	2	1	2	3	4	3 *	4	3	2	3	4	44

Note: a vertex 11 away from the initial vertex may be reached by the backdoor route of 32-25-18-11, for a total of 3 steps rather than the greedy algorithm's 5 steps. This yields a true diameter of 4, and a lower average diameter when compared to their algorithmic counterparts.

required is 4 with the "backdoor" routes allowed and 5 without them, we get a true diameter of 4 and an algorithmic (greedy) diameter of 5. These values are reflected in Table 6.5.1. ///

Example 6.5.4: Z_{32} using $S=\{1,6\}$.

From Appendix E, landing on vertices ±1 through ±15 again contributes 2·42=84 steps if the greedy algorithm is used or 2·40=80 if "backdoor" routes (to reach 14 and 15) are allowed. Reaching vertex 16 takes 5 jumps in either case, for an average diameter of 85/32=2.66 and an average (greedy) algorithmic diameter of 89/32=2.78. Observing that the maximum number of steps required is 5 with or without the "backdoor" routes, we get a diameter of 5 and an algorithmic (greedy) diameter of 5. ///

Example 6.5.5: Z_{32} using $S=\{1,5\}$.

Landing on vertices ±1 through ±15 contribute 2·42=84, plus 4 to reach 16, for an average diameter (no backdoor routes are of help) of 88/32=2.75. The diameter is 5. ///

Example 6.5.6: Z_{32} using S={1,4}.

Landing on vertices ±1 through ±15 contribute 2·44=88, plus 4 to reach 16, for an average diameter (algorithmic or not) of 92/32=2.88. The diameter is 5. ///

Example 6.5.7: Z_{32} using S={1,4,16}.

Since 4 and 16 divide 32 evenly, no backdoor routes are possible. Thus, landing on vertices ±1 through ±15 contribute 2·36=72 steps. Jumping to vertex 16 is only one step, while staying at zero obviously contributes zero steps to the total. Hence, we have an average algorithmic diameter of 73/32=2.28. Observing that the maximum number of steps to reach any vertex is 4, we have an algorithmic diameter of 4. ///

6.6 More Complicated Cayley Graphs, with Larger n

Logically, if using the truncated square root of n was a reasonable starting point for 1's partner in the generating set for 4-regular graphs, using the truncated cube root of n and the square of that value (i.e., the truncated 2/3rd root of n) as the partners of 1 would make a good starting point for a 6-regular graph. Similar reasoning should also work for larger values of k, thus providing an approach to explore more complicated graphs. In addition, there is hope that analysis of such graphs could be broken down into steps. If the problem may be reduced to examining blocks of vertices equivalent to smaller graphs with generating sets of size 4, then the work done in Chapter 6.5 could be used to finish the process.

Fortunately, we find that this is the case. Table 6.6.1 shows the results of this analysis for a number of examples on Z_{1024} and compares them with the figures for the hypercube and Schibell and Stafford's [SS] best result. (See Table 6.1.1.)

TABLE 6.6.1: COMPARISON OF DEGREES AND VARIOUS DIAMETERS FOR THE HYPERCUBE, SS3, AND SEVERAL CANDIDATE Z_{1024} GRAPHS

Graph Design and Generating Set	Degree	Diameter	Average Diameter	Algor. Diameter	Avg. Alg Diameter
Hypercube ($1024 = 2^{10}$)	10	10	5.0	10	5.0
SS3 Subgroup of S_{16} [SS]	6	7	4.5	unknown	unknown
Z_{1024} S={1, 4, 16, 64, 256}	10	≤8	≤4.64	8	4.64
Z_{1024} S={1, 7, 32, 224}**	8	≤8	≤5.25	10	5.5
Z_{1024} S={1, 6, 36, 216}	8	≤9	≤5.41	9	5.41
Z_{1024} S={1, 8, 64, 512}	7	≤11	≤6.37	11	6.37
Z_{1024} S={1, 12, 128}	6	≤14	≤7.52	14	7.52
Z_{1024} S={1, 10, 100}	6	≤14	≤7.46	14	7.46

** The diameter and average diameter presented here are upper bounds using backdoor routing that yield an improvement over the values obtained with the greedy algorithm shown in the last two columns. For each Z_{1024} considered, it is possible that the true and average diameters are smaller than the algorithmic, but these have generally not been pursued.

Presented below are the calculations for each of the entries in Table 6.6.1 based on Z_{1024}. Note that the true diameter and average diameter are not presented for most of these, the exception being for S={1,7,32,224}, where allowing backdoor routing showed an improvement on the greedy algorithm results. For those graphs whose generating sets are all powers of 4, it is unlikely that the true diameters are better than the algorithmic, since no backdoor routing will help. For the others, it is possible that an improvement over the algorithmic numbers exists, but no effort was made to find such values.

Example 6.6.2: Z_{1024} using S={1,8,64,512}.

The task in this more complicated graph is to determine how the greedy algorithm will get from zero to a particular multiple of 64, and from there to a target vertex. Each multiple of 64 may be considered a "landing pad", from which its (roughly) nearest 32 vertices in each direction may be most efficiently reached. In essence, if we can figure the weighted average of getting to each landing pad, the problem for figuring the number of steps from a landing pad becomes equivalent to the problem of finding the values for a graph of Z_{64} with generating set S={±1,±8}. This may be accomplished just as in Chapter 6.5.

Since the greedy algorithm works by getting as close as possible with each jump, some of these landing pad multiples of 64 will be preferred over others. For example, to reach vertex 480, jumping to vertex 512 (1 jump of 512) and then heading back toward zero 32 vertices (4 jumps of -8) is clearly faster than moving to 448 (7 jumps of 64) and then going 32 more vertices (4 jumps of +8) clockwise. Jumping to 512 and then going back four jumps of -8 is also faster than jumping to 512, then to 448 (1 jump of -64), then moving clockwise four jumps of +8. In other words, though 480 is equidistant from each of the potential landing pads 512 and 448, passing through 448 is less efficient, and the greedy algorithm will not bother to land there. In this sense, 512 is the preferred landing pad for 65 vertices (itself, and the 32 on each side) while 448 is preferred for only 64 vertices (itself, the 32 vertices before it, and the 31 vertices after it). We call landing pads such as 512 "master landing pads", since the intervals or domains of vertices that they control are as large as possible.

On the flip side of this analysis, some landing pads will be the landing pads of last resort, with the landing pads on both sides of it being the preferred way to reach the equidistant vertices between them. We call these "minor landing pads". Then consider the following multiples of 64:

0, 512:

Clearly, these are the master vertices for multiples of 64. Each will have 65 vertices in its domain, with an average distance from the master vertex of 256/65 = 3.94. The average to reach a master vertex is .5 steps, for a total average of 4.44 for 2·65 = 130 such vertices.

±256:

These are the minor landing pads. Each will have a domain of 63 vertices with an average distance from the landing pad of 248/63 = 3.94. The average to reach one of these vertices is 4, for a total average of 7.94 for 126 vertices.

Remaining vertices:

There are 12 standard landing pads, each with a domain of 64 vertices an average of 252/64 = 3.94 steps away. The average to reach one of these remaining vertices is 15/6 = 2.5 steps, for a total average of 6.44 steps to reach 12·64 = 768 vertices.

Thus, we get a weighted average of

$$[4.44\cdot130 + 7.94\cdot126 + 6.44\cdot768]/1024 = 6.37.$$

Observing that the maximum number of steps to reach any multiple of 64 is 4, and the maximum to reach a vertex no more than 32 away is 7, we obtain an algorithmic diameter of 11.

Note that simply assuming that every domain is equivalent in weight and simply calculating an average of [2·19 + 1]/16 = 2.44 steps to get to any particular multiple of 64 and 3.94 steps thereafter, we arrive at an overall average of 2.44+3.94 = 6.38 steps to any given vertex. This is certainly adequate as an estimate whenever our numbers go into n evenly and the size of each landing pad's domain is relatively large. ///

Example 6.6.3: Z_{1024} using $S=\{1,7,32,224\}$

Assume all landing pads that are multiples of 32 are of equal weight. Then this amounts to two levels of evaluating Z_{32} using $S=\{1,7\}$.

1. Sum the steps for reaching landing pads that are between 1 and 15 multiples of 32, and double this total.
2. Add the number of steps required to reach 512 to this total. (Reaching zero does not require any steps, and so does not contribute to this total.)
3. divide the total by 32 (the number of possible landing pads).

Step 1: sum is 42, doubled is 84.

Step 2: 4 steps are required to reach 512, so the new total is 88.

Step 3: divide by 32, yielding an average # of steps to reach the proper landing pad of 88/32 = 2.75.

From the proper landing pad, the target vertex is equally likely to be any in its domain, so that we need to perform a similar summation on the average number of steps needed to reach the 16 vertices further from zero and the 15 nearer zero, plus considering the landing pad itself. Going to the column labeled "1,7" in Table E.1, Appendix E, we must sum the entries in rows 1 through 15, double that total, add the entry in row 16, and divide by 32. This yields the average number of steps required to finish the journey from a landing pad to the target vertex. This process is, of course, the same as Steps 1 through 3 above, so that we again achieve an average of 2.75.

Hence, using a greedy algorithm, this set gives us an average (algorithmic) diameter of 5.5 and a maximum (algorithmic) diameter of 10, since trying to reach vertex 11 takes 5 steps in each table. However, if we use the algorithm that allows the "backdoor" route to reach ±11, then our sum is 40, doubled to 80, plus 4 is 84, divided by 32 is 2.625, so we obtain an average diameter of 5.25 and a maximum diameter of 8, as shown in Table 6.6.1. Unfortunately, the routing used to achieve these numbers requires a higher level of sophistication than the greedy algorithm, and may be difficult to implement. ///

Example 6.6.4: Z_{1024} using $S=\{1,6,36,216\}$.

Again envisioning the Cayley graph as a circle of vertices with appropriate labels and connecting edges, we can see that we must consider the average and frequency of occurrence of the following "domains":

The domain of the 0th multiple.

The domains of the ±1 through ±13th multiples of 36.

The domain of the 14th multiple of 36.

The domain of the -14th multiple of 36.

0: This domain contains 37 vertices with an average of 108/37 = 2.92 steps required from 0. Hence, the total average is 0+2.92 = 2.92 steps for these 37 vertices.

1-13:

These 26 domains each contain 36 vertices with an average of 105/36 = 2.92 steps required from its landing pad. The average number of steps to reach each base (arrived at by adding rows 1 through 13 of column "36, 216" in Table E.2, and dividing by 13) is 33/13 = 2.54, yielding an average of 2.54 + 2.92 = 5.46 for these 26·36 = 936 vertices.

14: This domain contains 24 vertices, with an average number of steps from 504 of 68/24 = 2.83, while it takes 4 steps to reach 504, yielding an average of 6.83 for these 24 vertices.

-14: This domain contains 23 vertices, with an average from -504 of 65/23 = 2.83 steps, while it also takes 4 steps to reach -504, yielding an average of 6.83 for these 23 vertices.

Hence, the weighted average for this generating set is:

$$2.92\cdot(37/1024) + 5.46\cdot(936/1024) + 6.83\cdot([24+23]/1024)$$
$$= 5.41.$$

Observing that the maximum number of steps to reach a multiple of 36 is 4 and the maximum number of steps to reach a vertex up to 18 vertices away (using 1 and 6) is 5, we obtain a maximum (algorithmic) diameter of 9. ///

Example 6.6.5: Z_{1024} using $S=\{1,4,16,64,256\}$.

Consider the following domains of multiples of 16:

0: 17 vertices with 32/17 = 1.88 steps from 0 on average.

1-31:

62 sets of 16 verts with an average of 30/16 = 1.875 steps from a landing pad. The average number of steps from a landing pad will be 88/31 = 2.839, yielding an average of 4.71 steps for 62·16 = 992 vertices.

32: 15 vertices with an average of 14/15 = .933 steps from 32·16 = 512, plus two steps to 512, for an average of 2.933 steps for these 15 vertices.

Then the weighted average is

$$[1.88 \cdot 17 + 4.71 \cdot 992 + 2.93 \cdot 15]/1024 = 4.64.$$

Again, note that we could have considered it as Z_{64} using $S=\{1,4,16\}$ to get to the nearest landing pad of 16, with each pad given equal weight. Reaching the desired multiple of 16 would require $[2 \cdot 88 + 2]/64 = 2.78$ steps on average, with the vertices in the domain of each multiple of 16 being reached in an average of $[2 \cdot 14+2]/16 = 1.875$ steps.

Hence, this would give an overall approximation of $2.78+1.875 = 4.66$ steps to a desired vertex. This matches quite well with the weighted average above.

Observing that the maximum number of steps to reach a multiple of 16 is 5 and the maximum number of steps to reach a vertex up to 8 vertices away (using 1 and 4) is 3, we obtain a maximum (algorithmic) diameter of 8. ///

Example 6.6.6: Z_{1024} using $S=\{1,12,128\}$.

Consider the following domains of multiples of 128:

0: 129 vertices with an average of $720/129 = 5.58$ steps from 0.

1-3 6 sets of 128 vertices with an average of $711/128 = 5.55$ steps from a landing pad. The average number of steps to a landing pad will be 2, for an average of 7.52 steps for $6\cdot128 = 768$ vertices.

4: This lands on 512, so its domain is 127 vertices, with an average of 702/127 = 5.53 steps from 512. It requires 4 steps to reach 512, so we get an average of 9.53 for these vertices.

Then the weighted average for this generating set is

$$(5.58\cdot129 + 7.52\cdot768 + 9.53\cdot127)/1024 = 7.52.$$

Observing that the maximum number of steps to reach a multiple of 128 is 4 and the maximum number of steps to reach a vertex up to 64 vertices away (using 1 and 12) is 10, we obtain a maximum (algorithmic) diameter of 14. ///

Example 6.6.7: Z_{1024} using $S=\{1,10,100\}$.

Consider the following domains of multiples of 100:

0-±4:

Each of these landing pads we can consider to have 100 vertices in its domain, with each vertex an average of [2·245 + 5]/100 = 4.95 steps from its landing pad. The average to a landing pad is 20/9 = 2.22 steps, for an overall average of 7.17 steps for 900 vertices.

+5: This lands on 500, so we have a domain of 50+12 = 62 vertices, with an average of (250+35)/62 = 4.6 steps from 500. Since it takes 5 to reach 500, we have an average of 9.6 for these vertices.

-5: We have a domain of 61 vertices, with an average of (250+32)/61 = 4.62 steps from -500, thus yielding 9.62 on average for these vertices.

Then the weighted average for this generating set is

$$(7.17\cdot 900 + 9.6\cdot 62 + 9.62\cdot 61)/1024 = 7.46.$$

Observing that the maximum number of steps to reach a multiple of 100 is 5 and the maximum number of steps to reach a vertex up to 50 vertices away (using 1 and 10) is 9, we obtain a maximum (algorithmic) diameter of 14. ///

Example 6.6.8: Z_{1024} using $S=\{1,8,64\}$.

Though this result is not listed in Table 6.6.1, we note that a similar analysis yields an average algorithmic diameter of 7.93 and an algorithmic diameter of 15. ///

Finally, we also present the results of analysis for several graphs based on Z_{4096} as compared with the hypercube, since this is the size of hypercube used in the Thinking Machine. (See [H1] and [H2] for details.) These results are listed in Table 6.6.9.

TABLE 6.6.9: COMPARISON OF DEGREES AND VARIOUS DIAMETERS FOR THE HYPERCUBE AND SEVERAL CANDIDATE Z_{4096} GRAPHS

Graph Design and Generating Set	Degr.	Diam.	Average Diameter	Algor. Diameter	Average Algor. Diam.
Hypercube (4096 = 2^{12})	12	12	6.0	12	6.0
S={1, 4, 16, 64, 256, 1024}	12	≤10	≤5.56	10	5.56
S={1, 8, 64, 512, 2048}	9	≤13	≤7.27	13	7.27
S={1, 8, 64, 512}	8	≤14	≤7.83	14	7.83
S={1, 10, 100, 1000}	8	≤15	≤8.03	15	8.03

Example 6.6.10: Z_{4096} using S={1,4,16,64,256,1024}.

This can be broken down into two levels, each level will be the equivalent of evaluating Z_{64} using S={1,4,16} as the generating set. Hence, consider the totals for {1,4,16} over the first 16 vertices in Table E.2, this being a sum of 37. Over the next 16 vertices, each entry will be one higher so the sum of the next sixteen entries would be 37+16 = 53. Hence, for vertices 1 through 31, we get a sum of 53-2+37 = 88. Double this to cover ±, we get 176, add 2 for landing on vertex 32, and we get an average of 178/64 = 2.78 steps per vertex.

It will be a close approximation to assume that the need for reaching each multiple of 64 in the Cayley graph of Z_{4096} is equal, thereby allowing us to state that it requires an average of $2 \cdot 2.78 = 5.56$ steps to reach any vertex desired. Since the maximum number of steps to reach any desired vertex in Z_{64} using $S=\{1,4,16\}$ is 5, the maximum algorithmic diameter of this Cayley graph is $2 \cdot 5 = 10$. ///

Example 6.6.11: Z_{4096} using $S=\{1,8,64,512,2048\}$.

Consider the domains of the following multiples of 64:

$0,\pm 8,\pm 16,\pm 24,32$:

Here, we have 8 master domains of 65 vertices with an average of $256/65 = 3.94$ steps from the landing pad, with an average of $11/8 = 1.375$ steps to the proper landing pad This yields an average of 5.32 steps for 520 vertices.

±4, ±12, ±20, ±28:

Without loss of generality, we may assume that these are the toughest multiples of 64 to get to, and hence the minor landing pads. We have 8 sets of 63 vertices, an average of 248/63 = 3.94 steps from the nearest multiple of 64. The average number of steps to one of these multiples of 64 is 20/4 = 5, yielding an average of 8.94 steps for 8·63 = 504 vertices.

remaining:

The remaining vertices all have domains that are fed by master vertices and themselves feed other landing pad vertices (possibly minor ones). We thus have 48 sets of 64 vertices an average of 252/64 = 3.94 steps from the multiple of 64, with an average of 81/24 = 3.375 steps to get to the proper landing pad yielding an average of 7.32 for 48·64 = 3072 vertices.

Thus, the weighted average is

$$[520\cdot 5.32 + 8.94\cdot 504 + 7.32\cdot 3072]/4096 = 7.27.$$

Observing that the maximum number of steps to reach a multiple of 64 is 6 (to reach $20\cdot 64 = 1280$) and the maximum number of steps to reach a vertex up to 31 (or 32; in this case they are the same for trying to reach 28 or 29 away) vertices away (using 1 and 8) is 7, we obtain a maximum (algorithmic) diameter of $6+7 = 13$. The degree is 9 because 2048 is its own inverse in Z_{4096}, while each of the others yields a distinct inverse under closure. ///

Example 6.6.12: Z_{4096} Using $S=\{1,10,100,1000\}$.

Consider the following domains of multiples of 100:

0: Zero is clearly a master vertex with a domain of 101 vertices, an average of $500/101 = 4.95$ steps from zero.

±10: These multiples are both master vertices (reached by one jump of 1000) with domains of 101 vertices, an average of 500/101 = 4.95 steps from ±1000. It is only one step to either of these master vertices, yielding an average of 5.95 for 202 vertices.

±6, ±16:

These multiples may be viewed as minor landing pads. (Without loss of generality, we may have chosen ±5 and ±15 to serve in their stead.) Each has a domain of 99 vertices an average of (70+110+65)·2/99 = 4.95 steps from the landing pad vertex. The average number of steps to these end-of-the-road vertices is 5.5, for an average of 10.45 steps to 4·99 = 396 vertices.

±20: Each of these multiples is a master vertex, but each has a domain consisting of 50 vertices back toward zero, itself, and 48 (for one, 47 for the other) towards 2048. Hence, 99+98 = 197 vertices an average of [250+239+250+232]/197 = 4.93 steps from its master vertex. With two steps to reach either ±2000, we have an average of 6.93 steps for 197 vertices.

Remaining landing pad vertices (1-5, 7-9, 11-15, 17-19):

Each of these 32 vertices contains 100 vertices in its domain, an average of (250+245)/100 = 4.95 steps from the landing pad. It takes an average of (70-14)/16 = 3.5 steps to reach these remaining landing pads, yielding an average of 8.45 steps for 3200 vertices.

Thus, we have a weighted average of

$$[101\cdot 4.95 + 202\cdot 5.95 + 396\cdot 10.45 + 8.45\cdot 3200]/4096 = 8.03.$$

Observing that it takes at most 6 steps to reach a landing pad and at most 9 steps to reach a vertex up to 50 away (using 1 and 10) we get an algorithmic diamter of 15. ///

Example 6.6.13: Z_{4096} using $S=\{1,8,64,512\}$.

Consider the following domains of multiples of 64:

Any multiple of 512 (0, ±1, ±2, ±3, 4):

Clearly, any multiple of 512 will be a best intermediate landing pad vertex, and hence a master vertex. Each has a domain consisting of the vertex itself, plus 32 vertices on either side. Hence, we have 8 sets of 65 vertices with an average of 256/65 = 3.94 steps from landing pad, with an average of 16/8 = 2 steps to the proper landing pad. This yields an average of 5.94 steps for 8·65 = 520 vertices.

±4, ±12, ±20, ±28 multiples of 64:

Without loss of generality, these are the toughest multiples to get to, thus being the minor landing pad vertices. We have 8 sets of 63 vertices, an average of 248/63 = 3.94 steps from the nearest multiple of 64. The average number of steps to one of these multiples of 64 is 22/4 = 5.5, yielding an average of 9.44 steps for 8·63 = 504 vertices.

remaining:

> The remaining are 48 sets of 64 vertices (32 toward lesser landing pad vertices, the vertex itself, and 31 vertices back in the direction of a higher order landing pad -- possibly a master vertex) an average of 252/64 = 3.94 steps from the multiple of 64, with an average of 96/24 = 4 steps to get to the proper landing pad yielding an average of 7.94 steps for 48·64 = 3072 vertices.

Thus, the weighted average is

$$[520 \cdot 5.94 + 9.44 \cdot 504 + 7.94 \cdot 3072]/4096 = 7.83.$$

Observing that the maximum number of steps to reach a multiple of 64 is 7 (to reach 28·64 = 1792) and the maximum number of steps to reach a vertex up to 31 (or 32; in this case they are the same for trying to reach 28 or 29 away) vertices away (using 1 and 8) is 7, we obtain a maximum (algorithmic) diameter of 7+7 = 14. ///

6.7 Network Construction and Routing Issues

Clearly, we have identified proposed graphs based on Z_n which show superior values to those of the hypercube. For example, the graph based on Z_{1024} with generating set $S=\{1,4,16,64,256\}$ has degree matching the hypercube on $n=1024$ vertices, yet has an algorithmic diameter of 8 (20% smaller than the hypercube's value of 10) and an average algorithmic diameter of 4.64 (an 8% reduction over the hypercube's value of 5.0). The flexibility in design shows that we can optimize in favor of either degree or diameter, depending on what is desired. In addition, any size graph may be used as a model, and a small amount of subsequent effort can identify an appropriate generating set.

For large values of n, physical construction of such a network should not be too difficult, since the circle of vertices may "accordion" into a donut shape, with "hinges" falling at the landing pad vertices and connections at the inside of the donut hole. Adding or subtracting vertices should not require too much rewiring, as the vertices are sequential, though jumps across the additions would need to be adjusted.

In short, only the concern about the possibility of routing bottlenecks need still be addressed. In the 1024 hypercube, routing is trivially determined by a 10-tuple with a 1 in any entry that requires a step along a corresponding edge. Travel along any of these identified vertices may be performed at any time during the routing with equal results, and hence the routing is bottleneck-free. Schibell and Stafford [SS] go to great pains to describe alternate routing methods in case of bottlenecks, but the process seems quite cumbersome and would clearly contribute to increasing the algorithmic diameters of their proposed networks.

One disadvantage of our greedy algorithm is that it is more likely to require a first step on a "large" generator rather than a "small" one, since a larger fraction of the possible target vertices are typically beyond the largest generator's reach than within it. This may cause some bottlenecking, depending on how the routing and processing timing is performed. However, it may not be too difficult to predetermine all the required jumps which would be produced by the steps above, and to select them in any order. Making a jump to another vertex would then require taking that "list" of jumps along, and crossing off each as it is performed.

Better still, though, is that any generating set consisting entirely of powers of a single generating element would allow an algorithm very similar to that used in the hypercube. For example, consider the graph based on the group Z_{1024}, with generating set $S=\{1,4,16,64,256\}$. Labeling all the vertices in base two would give us 10-tuples again, while the generators would also be 10-tuples, each consisting of one 1 and the rest zeros when written in base 2. The routing problem is thus nearly reduced to that of the hypercube's: eliminating 1's from the 10-tuple representing the difference between the current vertex and the target vertex. It would thus be a simple matter to determine which of the generators would contribute to that elimination process.

Labeling each vertex in base 4 might be even better, yielding a tidy generating set of ± versions of (0,0,0,0,1), (0,0,0,1,0), (0,0,1,0,0), (0,1,0,0,0), and (1,0,0,0,0), with the need to match the entries of the target route, rounding up where required by the greedy algorithm. For example, consider Z_{16} with generating set S={±1,±4}. This may be seen as having the generators ±(0,1) and ±(1,0). For a target route of, say, (1,2) (equivalent to +6 vertices away), the routing would obviously be to add two of the last entry (0,1) and one of the first entry (1,0). For a target route of (1,3), the second entry would be considered closer to four (3 rounds up) so we would see this target route as (2,-1), and this clearly requires 2 of (1,0) and 1 of -(0,1).

In either case, the routing is certainly straightforward and bottleneck-free. This routing and labeling concept explains why we put such an emphasis in Tables 6.6.1 and 6.6.9 on generating sets consisting of powers of a single number.

6.8 Comparison of Diameters to Theoretical Optimums

It is of some interest to compare the results of the above efforts to generate useful Cayley graphs with the theoretical optimums possible. Upper bounds remain in the realm of Theorems 2.2.4 and 2.3.1. These bounds were found to be quite soft, but there is no better alternative at this time. However, we may construct theoretical lower bounds on Cayley graph diameters based on both non-abelian and abelian groups. These lower bounds will be established using the following combinatorial arguments.

Clearly, if starting from a particular vertex in a Cayley graph of degree k on n vertices, no more vertices may be reached in d steps than may be reached in the same number of steps in an infinite k-regular tree. (This is because a tree never repeats a vertex, while a Cayley graph will eventually repeat vertices.) Hence, the distance from a vertex in an infinite k-regular tree to the nearest n vertices in the tree will be the smallest possible diameter for a Cayley graph on n vertices.

Because of the branch-back aspect of Cayley graphs, not all generators are available at any given position to seek out new vertices. Hence, we may refine this general argument on the infinite k-regular tree to those of rooted trees with specific numbers of branches available at each generation, depending on whether the underlying group is abelian or non-abelian.

Case 1: Non-abelian groups.

For a non-abelian group, the Cayley graph of degree k may be simulated by the tree beginning with a root with k branches extending from it. Each vertex at the end of a branch will then have k-1 possible branches extending from it to a new vertex. (The k^{th} branch would correspond to tracing back along the edge just arrived on, thus repeating a vertex.) At each new level, this process is repeated, so that, for $\rho \geq 2$, we have the following number of vertices in ρ generations below the root:

$$V_\rho = 1 + k + k(k-1) + k(k-1)^2 + \ldots + k(k-1)^{\rho-1}$$

Then, for a degree-6 Cayley graph on 1024 vertices, it would be impossible to do better than the smallest ρ for which $V_\rho \geq 1024$. From Table 6.8.1, ρ is 5, and so the diameter of such a Cayley graph must be at least 5. By this standard, Schibell's SS3 on 1024 vertices with degree 6 and diameter 7 is a pretty good attempt.

TABLE 6.8.1: Vρ FOR NON-ABELIAN TREES OF VARIOUS DEGREES (k) AND ρ GENERATIONS BELOW THE ROOT

	$\rho = 2$	$\rho = 3$	$\rho = 4$	$\rho = 5$	$\rho = 6$	$\rho = 7$
$k = 4$	17	53	161	485	1457	4373
$k = 5$	26	106	426	1706	6826	27306
$k = 6$	37	187	937	4687	23437	
$k = 7$	50	400	2500	15100	90700	
$k = 8$	65	585	4225	29705		
$k = 10$	101	911	8201	73811		
$k = 12$	145	1597	17569	not found		

Case 2: Abelian groups (e.g., the Z_n's).

Finding the number of vertices for a tree emulating an abelian group takes a bit more work, since the path $a*b*a$ will lead to the same vertex as the path $a*a*b$. Hence, the task is to construct a tree that, for each new generation, has no path that contains the same number of each of the same generators. In addition, the path $a*b*a^{-1}*b*a$ is the same as the path $b*b*a$, since one "a" and its inverse "a^{-1}" will cancel each other. Hence, we must count the number of paths without matching pairs of a generator and its inverse. The analysis of such trees is a combinatorial problem which is best broken down into the counting of distinct sets with no inverse pairs.

Degree is 4, with $S'=\{a,b,a^{-1},b^{-1}\}$:

Consider for this generating set how many distinct elements a path of generators may contain. Since there are only 2 generators and their inverse partners, we can only have paths that are either:

1. A sequence of only one generator. since there are only four distinct generators, there will be 4 such paths for each generation value, or a total of 4ρ of them.

2. A sequence of exactly two distinct generators. Since the group is abelian, the order of these generators is not important, but rather only the number of each of the two generators distinguishes it from any other path containing these two generators. That is, a*b*a is not distinct from a*a*b, since each sequence contains two a's and one b; however, a*a*a*b is distinct from a*a*b since the first sequence contains 3 a's while the second contains only 2 a's. Clearly, there are C(4,2)=6 ways to pair up the four generators, where two of these must be discarded since they contain inverse pairs. Thus, we have a total of 4 such sequences to consider.

Our task is to consider one such allowed pair of generators and to count how many distinct sequences we can construct with them of length 2 through ρ, where order is not important. For the pair {a,b} and a sequence of length y, this amounts to counting all the ways to have x of generator a and (y-x) of generator b, with at least one of each. This is a classic "ice cream distribution" problem in combinatorics. It comes under the heading "Combination with

Unlimited Repetition", and can be found in [JT], pages 66 through 70.

For two distinct generators, this is the equivalent of having a row of y-2 empty cones (two of the initial y cones are filled with one each of ice cream style a and ice cream style b) and deciding where to put a dividing partition between, or at the end of the row of, cones. Each place you can put a divider yields a different unordered sequence of y-2. For example, if y is 5, we can have the partitions "a|b,b,b,b", or "a,a|b,b,b", etc. Clearly, there are y-1 choices for where to put the divider. (For larger sets, we will use the combinatorial value of C(y-1,s-1) for each set size s and for y=s through y=ρ. Hence, we have for two distinct generators C(y-1,2-1) = C(y-1,1) = y-1, as above.) Thus, for degree 4, our tree will have

$V_2 = 1 + 4\rho + 4[(2-1)] = 1+8+4 = 13$ vertices,

$V_3 = 1 + 4\rho + 4[(2-1) + (3-1)] = 25$ vertices,

$V_4 = 1 + 4\rho + 4[(2-1) + (3-1) + (4-1)] = 41$ vertices, or, in general,

$$V_\rho = 1 + 4\rho + 4[1 + \ldots + (\rho-1)] \text{ for } \rho\geq 2.$$

Thus, we can generate these by noting that

$$V_{\rho+1} = V_\rho + 4 + 4\rho$$

when the degree is 4. Then for an abelian Cayley graph of degree 4 with 32 vertices, we must have a diameter of at least 4. We have, in fact, accomplished this with $S=\{1,7\}$ allowing a backdoor route. With only the greedy algorithm, our best diameter is 5.

Degree is 6 with $S'=\{a,b,c,a^{-1},b^{-1},c^{-1}\}$:

A similar argument to that shown in the degree 4 case has

1. 1 root vertex.
2. 6ρ sequences consisting of only one generator and of

length 1 through ρ. (6 of each such length, since 6 distinct generators.)

3. Sequences of exactly two distinct generators. There are C(6,2)-(# of inverse pairs) = 12 such possible pairs of generators, each of which contributes $[1+2+...+(\rho-1)]$ distinct vertices (sequences) as reasoned above in the degree 4 case.

4. Sequences of exactly 3 distinct generators. There are $2^3=8$ such generating trios, since such a sequence must contain only (a's or a^{-1}'s) and (b's or b^{-1}'s) and (c's or c^{-1}'s). Each such trio will contribute C(y-1,2) distinct sequences for each value of y from 3 through ρ. ("Ice cream distribution" of choosing among y-1 places to put 2 dividers. For example, when y=7, three of these are taken up by distributing one each to a, b, and c, leaving 4 more choices. Then we have 4 more generators and 2 dividers left to position. This amounts to deciding where among 4+2 = 6 spots to place the two dividers, since that determines how many of each generator is in the sequence. Hence, we would have

$C(6,2) = C(7-1,2) = C(y-1,2)$ such sequences for $y=7$.)

Hence, we get for degree 6 that

$$V_\rho = 1 + 6\rho + 12[1+2+ \ldots + (\rho-1)]$$
$$+ 8[C(2,2) + C(3,2) + \ldots + C(\rho-1,2)]$$

and so

$$V_1 = 1+6 = 7$$

$$V_2 = 1+ 6\cdot2 + 12 = 25$$

$$V_3 = 1 + 6\cdot3 + 12[1+2] + 8[1] = 63$$

and

$$V_{\rho+1} = V_\rho + 6 + 12\rho + 8[C(\rho,2)] \text{ for } \rho\geq3.$$

Running through some of these, we get that $V_4=129$, $V_5=231$, $V_6=377$, $V_7=575$, $V_8=833$, $V_9=1159$. This tells us that the best we could do for an abelian group of size 1024 and degree 6 is a diameter of at least 9. In comparison, our algorithmic diameter is 14.

Degree is 8 with $S'=\{a,b,c,d,a^{-1},b^{-1},c^{-1},d^{-1}\}$:

The same arguments as above apply, though we need to modify the coefficients appropriately and to add a term for sequences of exactly 4 distinct generators. Hence we have

1. 1 root vertex.
2. 8ρ sequences consisting of only one generator and of length 1 through ρ,
3. $C(8,2)$-(4 inverse pairs) $= 8\cdot 7/2 - 4 = 24$ possible sets of generator pairs, each of which again contributes $[1+\ldots+(\rho-1)]$ distinct vertices (sequences).
4. $C(8,3)$-(# of sets with inverse pairs) $= 8\cdot 7\cdot 6/[3\cdot 2]$ - (4 pair choices)$\cdot$(6 partner choices) $= 56-24 = 32$ possible trios of exactly 3 distinct generators. Again, each trio contributes $C(y-1,2)$ distinct sequences for each value of y from 3 through ρ.
5. $2^4=16$ allowed sets of 4 distinct generators. Each "quad" contributes $C(y-4+3,3) = C(y-1,3)$ distinct sequences for each value of y from 4 through ρ.

Hence, we get for degree 8 that

$$V_\rho = 1 + 8\rho + 24[1+ \ldots +(\rho-1)] + 32[C(2,2)+ \ldots +C(\rho-1,2)]$$
$$+ 16[C(3,3)+ \ldots +C(\rho-1,3)]$$

and so

$$V_1 = 1+8 = 9$$

$$V_2 = 1 + 8\cdot2 + 24 = 41$$

$$V_3 = 1 + 8\cdot3 + 24[1+2] + 32[1] = 129$$

$$V_4 = 1 + 8\cdot4 + 24[1+2+3] + 32[1+C(3,2)] + 16[1] = 321$$

and

$$V_{\rho+1} = V_\rho + 8 + 24\rho + 32[C(\rho,2)] + 16[C(\rho,3)] \text{ for } \rho\geq4.$$

Running through some of these values we get V_5=681, V_6=1289, V_7=2241, V_8=3649, and V_9=5641. This tells us that the lower limit for an abelian group of degree 8 and size 1024 is diameter 6, while for degree 8 and size 4096 the lower limit on diameter is 9. Our algorithmic diameters are 9 (8 allowing "backdoor routes") and 14, respectively.

Degree is 10 with $S'=\{a,b,c,d,e,a^{-1},b^{-1},c^{-1},d^{-1},e^{-1}\}$:

The same type of argument leads to

$$V_\rho = 1 + 10\rho + 40[1+\ldots+(\rho-1)] + 80[C(2,2)+\ldots+C(\rho-1,2)]$$
$$+ A[C(3,3)+\ldots+C(\rho-1,3)] + 2^5[C(4,4)+\ldots+C(\rho-1,4)]$$

where

$$\begin{aligned} A &= \text{\# of allowed sets of 4 distinct generators} \\ &= C(10,4) - (\text{\# of sets with one or two distinct pairs}) \\ &= C(10,4) - [5C(8,2) - C(5,2)] \\ &= 210 - [5\cdot 28 - 10] = 210-130 = 80. \end{aligned}$$

Hence, for $\rho \geq 5$, we have

$$V_{\rho+1} = V_\rho + 10+ 40\rho + 80[C(\rho,2)] + 80[C(\rho,3)] + 32[C(\rho,4)].$$

Running through some totals, we get $V_1=11$, $V_2=61$, $V_3=231$, $V_4=681$, $V_5=1683$, $V_6=3653$, $V_7=7183$. Hence, the lower limit for an abelian group of degree 10 and size 1024 is diameter 5, while for degree 10 and size 4096 the lower limit on diameter is 7. We have only examined a group of size 1024 with degree 10, for which we found a diameter of 8.

Note that the previous analysis only considered generating sets of even size. It would not be difficult to include odd-sized generating sets (i.e., S contains the element n/2) as the odd element n/2 is its own inverse and therefore may only be used once in the sequence. However, this was not pursued, since we are primarily interested in even generating sets and the analysis is used only to get a rough idea of how "tree-like" the Z_n-based Cayley graphs are.

Note also that, when all the recursion relations are examined together, a clear trend emerges: each increase from ρ to ρ+1 increases the number of vertices in the tree by a polynomial in ρ of degree (k/2)-1. This is a direct result of the "ice cream distribution" evaluation. For a given sequence length ρ (the number of "cones") and a given subset of generators of size s, this evaluation yields a result of C(ρ-1,s-1). Hence, for ρ+1 and a maximum s=k/2 (inverse pairs are not allowed in the same sequence) the highest term will be a of degree (k/2)-1 in ρ. That is, for each degree k=2m, we have

$$V_{\rho+1} = V_\rho + a_0 + a_1\rho + \ldots + a_{m-1}\rho^{m-1},$$

for appropriate constants a_i.

In Table 6.8.2 we present these results, which generally average around 50% over the theoretical lower limit found with the previous combinatorial argument.

TABLE 6.8.2: THEORETICAL MINIMUM DIAMETERS FOR GRAPHS BASED ON ABELIAN GROUPS vs. DIAMETERS OF Z_n-BASED GRAPHS

Degree	4	6	8	10	8	10
n =	32	1024	1024	1024	4096	4096
Theoretical Bound	4	9	6	5	9	7
Z_n	5 (4*)	14	9	8	14	**
* Note that 4 is the diameter allowing backdoor routing. ** No such Z_n-based graph was studied.						

6.9 Conclusions and Unanswered Questions

From all the explorations in Chapter 6, it is clear that Cayley graphs based on the groups of form Z_n may be used to improve on the hypercube network model for parallel processing supercomputers. At each value of 2^k examined, the equivalent Z_n on k generators yields an algorithmic diameter that is roughly 20% smaller and an average algorithmic diameter that is on the order of 8% smaller. For n=1024, a reduction in generators by 20% (over the hypercube's 10) may also be accomplished with a 10% better algorithmic diameter, though at a sacrifice of roughly 8% in average algorithmic diameter. (Table 6.6.1.) With the "building block" concept of generating set construction, these trends should be valid for every possible value of 2^k.

In addition, the family of Cayley graphs based on Z_n has no restriction on values for n, so full flexibility in network size is available. Finally, routing for appropriate generating sets is simple and bottleneck-free.

It is not clear how to evaluate or weight all the trade-offs between the Cayley graphs on Z_n and those on some of the more exotic, theoretically superior groups mentioned in Table 6.1.1. Though [SS] admit to the somewhat inefficient nature of their general-purpose routing algorithm described in Chapter 5, it is possible that case-specific routing would significantly improve the performance of their candidate architectures listed in Table 6.1.1. It would be interesting to explore the underlying assumptions of Pitelli and Smitley's [PS] computer simulation on network performance, and to see how the equivalent Z_n's would compare in appropriate simulation to the candidate architectures proposed by Schibell and Stafford [SS] or other researchers.

The proposed Cayley graphs based on exotic groups were, in principal, designed to optimize the theoretical constants explored in earlier chapters, so it is reasonable to assume that those constants have better values than Cayley graphs based on Z_n. However, since we claim performance improvements over the hypercube with our family of Z_n, we should show at least roughly comparable values of λ_1 (and, therefore, expansion characteristics) to those of the hypercube for equivalent n.

Alon and Milman [AM] assert that $\lambda_1=2$ for every hypercube. We claim here that this value may be at least matched by the Z_n architecture for each $n=2^k$, where k is even. By the Lovasz algorithm in Appendix D, we have that the eigenvalues of the adjacency matrix A(G) on the Cayley graph based on Z_n with generating set $S=\{1,2^2,2^4,\ldots,2^{k-2}\}$ are

$$\xi_t = 2[\cos(2\pi t/n) + \cos(2\cdot 2^2\pi t/n) +\ldots+ \cos(2\cdot 2^{k-2}\pi t/n)]$$
$$= 2[\cos(2\pi t/n) + \cos(8\pi t/n) +\ldots+ \cos(\pi t/8) + \cos(\pi t/2)]$$

for t from 1 through n-1. When $t=2^{k-2}$, the corresponding eigenvalue is k-2, since the first term in the parenthesis is zero while all other terms are equal to 1. Hence, $\lambda_1\leq 2$.

Clearly, the last term in the sum is zero whenever t is odd, and is -1 whenever t is a multiple of 2 but not of 4. Hence, only for $t\equiv 0 \pmod 4$ does the hope remain that $\xi_t>k-2$. However, this argument repeats itself when considering the second to last term in the summation. For t a multiple of 4, the second to last term will be zero or -1 for all such multiples except when $t\equiv0 \pmod{16}$; that is, for every fourth multiple of four. Obviously, this process of elimination continues until $t=2^{k-2}$, from which we originally obtained the value of $\xi_t=k-2$. Hence, $\xi_t\leq k-2$ for every t, implying that $\lambda_1\geq2$, and so $\lambda_1=2$.

So, we have constructed an infinite family of Cayley graphs built on Z_n with λ_1 always equal to 2. (It is possible that a better choice of generating sets exist for each n to improve on this number, but no efforts have been made to establish such a claim.) Note that this result forces all the theoretical constants explored in earlier chapters to be bounded away from zero for this family of graphs.

However, we have **not** constructed a Ramanujan family, since $\mu \geq k-\lambda_1 = k-2 > 2(k-1)^{1/2}$ for all $k\geq7$, yet our degree grows infinitely as $\log_2 n$. We did show, though, that if the degree is allowed to expand as $\log_2 n$, we can describe an infinite family of graphs with $\lambda_1=2$, clearly bounded away from zero. Observe that, for fixed positive integer r, there exists $m\in\mathbb{I}$ such that $\log_2 n < n/r$ for every $n\geq m$. Thus, by dropping all the graphs with n<m from the sequence described above, the remaining infinite (sub-) sequence forms an infinite family of graphs with $\lambda_1=2$ and degree less than n/r. Hence, we can construct an infinite family of graphs of degree less than any fixed fraction of n where $\lambda_1=2$, thus bounded away from zero.

Earlier, we showed that, as n gets very large, it is impossible to bound λ_1 (or other constants) away from zero with fixed degree of 4 and generating set $S=\{\pm1,\pm x\}$. Expanding on this, we speculate that if the degree is bounded above by any fixed number, then λ_1 will tend toward zero as n gets very large, regardless of the choice of generating sets. Certainly the lower bounds achieved in Theorem 3.4.3 go to zero. Also, it appears obvious that the Lovasz algorithm sends $max\{\xi_t\}$ to k as n gets very large, thus sending λ_1 to zero. (Intuitively, as n gets very large yet the number of terms in the summation remain fixed, there should be enough "gaps" in the generator values to allow a choice of t that sends each of the values of the cosine terms to very near 1.) Establishing an elegant proof (or refutation) of this speculation would be satisfying.

It is unclear if there is some n-related function that is a "break" point for restrictions on the degree of the Cayley graphs to construct an infinite family of graphs with λ_1 bounded away from zero. We have shown that it is possible if the degree is allowed to increase by $\log_2 n$, yet not possible for degree 4 (and, we speculate, for any fixed degree). It would be interesting to see if some function of n that is more restrictive than a log-based function would prevent such a family from being constructed.

In Section 6.4, we noted a trend (over most values of n examined) that at least one of the generating sets of size 4 produced a Cayley graph with simultaneously the highest λ_1 and the smallest diameter possible. This is worth further study, since establishing such a correlation for graphs of higher degree and larger n would be quite useful in finding generating sets (and, thus, the specific Cayley graphs) which optimize network diameters.

Short of establishing this correlation, or some other equivalently useful relationship, computer programs may be written to, for specific n, exhaustively examine all candidate Z_n-based Cayley graphs for minimum algorithmic diameter. This method would be crude and tedious, but would eventually supply a list of desirable candidates.

Though Kazhdan constants proved to yield only very loose bounds in λ_1, h(G), and c_{max}, some of the patterns observed while solving for Kazhdan numbers were of help. In particular, computer search results for best generating sets to maximize $\hat{\mathcal{K}}_r(S)$ helped establish the set $S=\{\pm1,\pm x\}$ (where x is the truncated square root of n) as a top candidate for producing graphs with good (algorithmic) diameter. A proof (or disproval) of the Chapter 3 speculation that $\{\pm1,\pm x\}$ produces as good or better a value for $\hat{\mathcal{K}}_r(S)$ than any other generating set of size 4 may provide further useful insights.

In conclusion, though Chapters 2 through 4 demonstrate that λ_1 is directly related to both the diameter and the expansion characteristics of the graph, the theoretical bounds found in the surveyed literature proved to be quite loose for Z_n-based Cayley graphs. These bounds may, in fact, be rather poor for most (all?) kinds of graphs, though exploring that topic is beyond the scope of this thesis.

On the more concrete side, this chapter's study of Cayley graphs based on underlying groups from the family of Z_n showed that they cannot compete with the true full and average diameters achieved by Cayley graphs based on more exotic underlying groups. However, the flexibility, simplicity, and efficient routing of Z_n-based Cayley graphs provide practical results superior to those of the hypercube, and hence may deserve a closer look from computer researchers as candidate models for parallel processor supercomputer architectures.

REFERENCE LIST

[A] Alon, N. "Eigenvalues and Expanders," Combinatorica, 6 (1986), 85-98.

[AM] Alon, N., and V. D. Milman. "λ_1, Isoperimetric Inequalities for Graphs, and Superconcentrators," Journal of Combinatorial Theory, Ser. B, Vol. 38, No. 1 (February, 1985), 73-88.

[BdlH] Bacher, Roland, and Pierre de la Harpe. "Exact Values of Kazhdan Constants for Some Finite Groups," Journal of Algebra, 163 (1994), 495-515.

[BMS] Biggs, N. L., Bojan Mohar, and John Shawe-Taylor. "The Spectral Radius of Infinite Graphs," Preprint Series, 24, No. 174 (1986), (Department of Mathematics, University E. K. Ljubljana).

[Ch] Chung, F. R. K. "Diameters and Eigenvalues," Journal of the American Mathematical Society, Vol. 2, No. 2 (April, 1989), 187-196.

[CL] Chartrand, Gary, and Linda Lesniak. Graphs & Digraphs. 2d ed. Monterey: Wadsworth & Brooks/Cole Advanced Books & Software, 1986.

[Dia] Diaconis, Persi. Group Representation in Probability and Statistics. Hayward: Institute of Math. Statistics, 1988.

[FIS] Friedberg, Stephen H., Arnold J. Insel, and Lawrence E. Spence. Linear Algebra. 2d ed. Englewood Cliffs: Prentice-Hall, 1989.

[H1] Hillis, W. Daniel. "The Connection Machine," Scientific American, Vol. 256 (June, 1987) 108-115.

[H2] Hillis, W. Daniel. The Connection Machine. Cambridge: MIT Press, 1989.

[H] Hungerford, Thomas W. Algebra. New York: Springer-Verlag, 1974.

[JT] Jackson, Bradley W., and Dmitri Thoro. Applied Combinatorics with Problem Solving. Reading: Addison-Wesley, 1990.

[Lov] Lovasz, L. "Spectra of Graphs with Transitive Groups," Periodica Mathimatica Hungarica, Vol. 6 (2) (1975), 191-195.

[LPS} Lubotzky, A., R. Phillips, and P. Sarnak. "Ramanujan Graphs," Combinatorica, 8 (1988), 261-277.

[Lub] Lubotzky, Alexander. Discrete Groups, Expanding Graphs and Invariant Measures. Progress in Mathematics, Vol. 125. Basel: Birkhauser-Verlag, 1994.

[PS] Pittelli, F., and D. Smitley. "Analysis of a 3D Toroidal Network for a Shared Memeory Architecture," Proceedings of Supercomputing 88 Conference, (1988), 42-47.

[Rob] Roberts, Fred S. Applied Combinatorics. Englewood Cliffs: Prentice-Hall, 1984.

[Sch] Schellwat, Holger. "Highly Expanding Graphs Obtained from Dihedral Groups," DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol. 10 (1993), 117-123.

[SS] Schibell, Stephen T., and Richard M. Stafford. "Processor Interconnection Networks from Cayley Graphs," Discrete Applied Mathematics, 40 (1992), 333-357.

APPENDIX A

RAYLEIGH QUOTIENT THEOREM

In Chapter 2, the proof of Theorem 2.2.1 requires a step that is justified here.

First, let B be an n×n self-adjoint matrix. (For real matrices, this requires that it be symmetric.) The Rayleigh quotient for $x \in \mathbb{C}^n$ (or $\mathbb{R}^n$) and $x \neq 0$ is defined as the scalar

$$R(x) = (Bx, x)/\|x\|^2.$$

Then we present a standard theorem which links this scalar to the eigenvalues of B.

THEOREM A.1 ([FIS], Theorem 6.36): For such a B we have

$$\begin{aligned} \mathit{max}R(x) &= \text{largest eigenvalue of } B \\ \mathit{min}R(x) &= \text{smallest eigenvalue of } B. \end{aligned}$$

***Proof*:** By several well-known theorems, it is possible to select an orthonormal basis $\beta=\{x_0,x_1,\ldots,x_{n-1}\}$ of eigenvectors of B such that $Bx_i = \lambda_i x_i$, where $\lambda_{n-1} \geq \lambda_{n-2} \geq \ldots \geq \lambda_1 \geq \lambda_0$ are real eigenvalues. Then, we may write x as a linear combination of the x_i's, as

$$x = \sum_{i=0}^{n-1} a_i x_i,$$

where $a_i \in \mathbb{C}$ (or $\mathbb{R}$) for every i. Hence,

$$\begin{aligned} R(x)\|x\|^2 &= (Bx,x) \\ &= (\sum_{i=0}^{n-1} a_i\lambda_i x_i,\ \sum_{j=0}^{n-1} a_j x_j) \\ &= \sum_{i=0}^{n-1} |a_i|^2\lambda_i\|x_i\|^2, \end{aligned}$$

since each term $x_i x_j = 0$ whenever $i \neq j$ because the x_i's form an orthonormal basis. Also due to orthonormality, $\|x_i\|^2=1$ for each value of i, so

$$R(x)\|x\|^2 = \sum_{i=0}^{n-1} |a_i|^2\lambda_i .$$

But

$$\sum_{i=0}^{n-1} |a_i|^2\lambda_0 \le \sum_{i=0}^{n-1} |a_i|^2\lambda_i \le \sum_{i=0}^{n-1} |a_i|^2\lambda_{n-1},$$

since $\lambda_0 \le \ldots \le \lambda_i \le \ldots \le \lambda_{n-1}$. Then, by moving λ_0 and λ_{n-1} outside of the summation signs in the previous inequality and noting that $\|x\|^2 = \sum_{i=0}^{n-1} |a_i|^2$, we have

$$\lambda_0 \sum_{i=0}^{n-1} |a_i|^2 = \lambda_0\|x\|^2 \le R(x)\|x\|^2 \le \lambda_{n-1}\|x\|^2 = \lambda_{n-1} \sum_{i=0}^{n-1} |a_i|^2 .$$

Hence

$$\lambda_0 \le R(x) \le \lambda_{n-1} . \qquad \square$$

Clearly, if we choose a vector (n-tuple) x which is in a subspace of $\mathbb{R}^n$ that is orthogonal to the eigenvector x_0, then it may be written as a sum of eigenvectors x_1 through x_{n-1}. That is

$$\begin{aligned} x &= 0x_0 + a_1x_1 + a_2x_2 + \ldots + a_{n-1}x_{n-1} \\ &= a_1x_1 + a_2x_2 + \ldots + a_{n-1}x_{n-1}. \end{aligned}$$

In the proof of Theorem 2.2.1, just such a vector f is created, where it may be written as a linear combination of the eigenvectors associated with the eigenvalues λ_1 through λ_{n-1}. That is, $a_0=0$ in this case. Then

$$R(f)\|f\|^2 = (Qf,f) = \sum_{i=0}^{n-1} |a_i|^2\lambda_i = \sum_{i=1}^{n-1} |a_i|^2\lambda_i,$$

and so

$$\lambda_1\|f\|^2 \leq R(f)\|f\|^2 \leq \lambda_{n-1}\|f\|^2.$$

But $R(f)\|f\|^2 = (Qf,f)$, so

$$\lambda_1\|f\|^2 \leq (Qf,f) \leq \lambda_{n-1}\|f\|^2$$

as desired in the proof of Theorem 2.2.1.

APPENDIX B

MAPLE PROGRAMS FOR LOVASZ AND KAZHDAN CONSTANTS PLUS SELECTED RESULTS

B.1 Kazhdan Constant Program

The following program KAZ1.MS, written in MapleV, evaluates Kazhdan constants for groups of the form Z_n. The range of groups Z_n is specified by placing the desired lower and upper values of the range (inclusive) in place of A and B in line 2 ("for n from A to B do"). It is constructed to find the best (highest) Kazhdan constant for each Z_n for the case of generating sets consisting of:

a. all elements of Z_n

b. any set of size 2 up through size "ztop" (inclusive) of any elements of Z_n.

Note that the search method is by exhaustion of all possible candidates for each size range (excluding inverses in the "second half" of Z_n's elements). Note also that any candidate set examined which is not actually a generating set for Z_n will be eliminated from consideration since it will produce a Kazhdan constant of zero.

KAZ1.MS

```
> with(numtheory):
> for n from A to B do
> lprint(n);
> ztop:=3:
> a:=n mod 2:
> if a=0 then t:=n/2:
> else t:=(n-1)/2:
> fi:
> g:=factorset(n);
> q:=nops(g):
>    if q=1 and a=0
>          then fall:=n/2;
>    elif isprime(n)
>          then fall:=(n-1)/2;
>    elif q>1 and a=0
>          then fall:=(g[2]-1)*n/(2*g[2]);
>    else
>          fall:=(g[1]-1)*n/(2*g[1])
>    fi;
> lprint(`f for S=n is`, fall);
> lprint(`The Kazhdan constant for the whole group is`,
>          evalf(2*sin(Pi*fall/n), 4));
> z:=2:
> x:=trunc(sqrt(n)):
```

```
> f:=x:
>
> while z<=ztop do
>    list1:=[seq(i, i=1..(z-1))]:
>    listf:=[f]:
>    lists:=[op(list1), op(listf)]:
>    bestlist:=lists:
>    m:=0:
>
>    while m<z do
>       if f=fall
>         then m:=z:
>       elif lists[z-m]<t-m
>         then
>            if m=0
>              then lists:=subsop(z-m=lists[z-m]+1,lists):
>            else
>              lists:=subsop(z-m=lists[z-m]+1,lists):
>              for j from 1 to m do
>                 lists:=subsop(z-m+j=lists[z-m]+j,lists):
>              od:
>            fi:
>            m:=0:
>            fofs:=t;
>            i:=1:
>            while i<=t do
>              fhold:=f:
>              for j from 1 to z do
>                 fhold:=max(fhold,abs(mods(i*lists[j],n))):
>              od:
>              if fhold=f then
>                 i:=t+1;
>              else
>                 fofs:=min(fofs,fhold);
>                 if i=t then
>                   bestlist:=lists;
>                   f:=fofs;
>                   i:=t+1;
>                 else i:=i+1;
>                 fi:
```

```
>              fi:
>            od:
>         else
>            m:=m+1:
>         fi:
>       od:
>
>       lprint(`The best f for S of size =`,z,`is`,f);
>       lprint(`The best set for S is`,bestlist);
>       lprint(`the Kazhdan constant is`,
>                       evalf(2*sin(Pi*f/n),4));
>       if f=fall
>         then z:=ztop+1;
>       else
>         z:=z+1:
>       fi:
>    od:
> od:
```

Selected results of running KAZ1.MS for n from 10 to 200 are presented below in Table B.1.1.

TABLE B.1.1: SELECTED RESULTS OF KAZ1.MS FOR n=10 THROUGH 200

	|S|=4			|S|=6			|S|=8			|S|=n	
n	f	S	$\hat{\kappa}_r$	f	S	$\hat{\kappa}_r$	f	S	$\hat{\kappa}_r$	f	$\hat{\kappa}_r$
10	3	1,3	1.62	4	1,2,4	1.90		**		4	1.90
11	3	1,3	1.51	4	1,2,4	1.82	4	1,2,3,4	1.82	5	1.98
12	3	1,3	1.41	4	1,2,4	1.73		**		4	1.73
13	3	1,3	1.33	4	1,2,4	1.65	5	1,2,3,5	1.87	6	1.99
14	3	1,3	1.25	5	1,3,5	1.80	6	1,2,3,7	1.95	6	1.95
15	3	1,3	1.18	5	1,2,5	1.73		**		5	1.73
16	4	1,4	1.42	5	1,2,5	1.66	8	1,2,4,8	2	8	2
17	4	1,4	1.35	5	1,2,5	1.60	7	1,2,4,8	1.92	8	1.99
18	4	1,4	1.29	6	1,2,6	1.73		**		6	1.73
19	4	1,4	1.23	6	1,2,6	1.67	7	1,2,4,7	1.83	9	1.99
20	4	1,4	1.18	6	1,2,6	1.62	8	1,2,4,8	1.90	8	1.90
21	4	1,4	1.13	6	1,2,6	1.56	7	1,2,3,7	1.73	7	1.73
22	4	1,4	1.08	6	1,2,6	1.51	8	1,2,4,8	1.82	10	1.98
23	4	1,4	1.04	6	1,2,6	1.46	8	1,2,4,8	1.78	11	1.99
24	4	1,4	1	6	1,2,6	1.41	8	1,2,3,8	1.73	8	1.73
25	5	1,5	1.18	7	1,3,7	1.54	10	1,2,5,10	1.90	10	1.90
26	5	1,5	1.14	8	1,3,9	1.65	10	1,2,5,10	1,87	12	1.99
27	5	1,5	1.10	9	1,3,9	1.73		**		9	1.73
28	5	1,5	1.06	8	1,3,10	1.56	10	1,2,5,10	1.80	12	1.95
29	5	1,5	1.03	8	1,3,8	1.53	10	1,2,5,10	1.77	14	2.00
30	5	1,5	1	9	1,3,9	1.62	10	1,2,4,10	1.73	10	1.73
31	5	1,5	.971	9	1,3,9	1.58	10	1,2,4,10	1.70	15	2.00
32	5	1,5	.943	9	1,3,10	1.55	10	1,2,4,10	1.66	16	2
33	5	1,5	.916	9	1,3,9	1.51	11	1,2,4,11	1.73	11	1.73

** |S|=8 column not required when f for |S|=6 is the same as f for all generators.

	|S|=4			|S|=6			|S|=8			|S |=n	
n	f	S	$\hat{\kappa}_r$	f	S	$\hat{\kappa}_r$	f	S	$\hat{\kappa}_r$	f	$\hat{\kappa}_r$
34	5	1,5	.892	9	1,3,9	1.48	11	1,2,4,11	1.70	16	1.99
35	5	1,5	.868	9	1,3,9	1.45	11	1,2,4,11	1.67	14	1.90
40	6	1,6	.908	10	1,3,10	1.41	12	1,2,4,12	1.62	16	1.90
45	6	1,6	.813	11	1,3,11	1.39	15	1,2,5,15	1.73	15	1.73
50	7	1,7	.852	12	1,3,12	1.37	15	1,2,5,15	1.62	20	1.90
55	7	1,7	.779	12	1,3,12	1.27	17	1,2,12,18	1.65	22	1.90
60	7	1,7	.728	13	1,4,13	1.28	18	1,2,20,26	1.64	29	1.73
65	8	1,8	.766	16	1,4,16	1.41		*		26	1.90
70	8	1,8	.703	16	1,4,17	1.32		*		28	1.90
75	8	1,8	.658	16	1,4,17	1.24		*		25	1.73
80	8	1,8	.618	16	1,4,16	1.18		*		32	1.90
85	9	1,9	.653	17	1,4,17	1.18		*		34	1.90
90	9	1,9	.618	18	1,4,18	1.18		*		30	1.73
95	9	1,9	.587	19	1,4,19	1.17		*		38	1.90
100	10	1,10	.618	20	1,4,20	1.18		*		40	1.90
107	10	1,10	.579	20	1,4,20	1.11		*		53	2.00
121	11	1,11	.563	22	1,5,22	1.08		*		55	1.98
133	11	1,11	.514	25	1,5,26	1.11		*		57	1.95
144	12	1,12	.518	26	1,39,50	1.08		*		48	1.73
152	12	1,12	.491	26	1,6,29	1.02		*		72	1.99
160	12	1,12	.467	27	1,6,29	1.01		*		64	1.90
169	13	1,13	.479	28	1,5,28	.995		*		78	1.99
178	13	1,13	.455	29	1,5,29	.979		*		88	2.00
184	13	1,13	.440	30	1,5,30	.980		*		88	2.00
196	14	1,14	.445	31	1,6,31	.954		*		84	1.95

* Note that calculations for |S|=8 were not made for larger n due to high time required.

B.2 Lovasz Eigenvalue Program

Below is a MapleV program called LOVSUM.MS which is designed to yield the best value of λ_1 and the lowest spectral radius μ for a given n during a search through all generating sets of the form $S=\{\pm1,\pm y\}$. The range of n is specified by giving values to A and B in the first line. Clearly, some modification in this program similar to KAZ1.MS in B.1 above would enable the search to exhaust all generating set possibilities, and therefore to find the best λ_1 and μ for generating sets of size 4. The results of such a modified program might yield higher λ_1 and lower μ in some cases, though, as speculated in Chapter 6.4, it seems likely that no improvement would be found for the generating sets of size 4. However, no such program was written because we need ±1 in the generating set to use the greedy algorithm, because the run time for KAZ1.MS was extensive, and because such a program would require checking each candidate set to see whether it would indeed generate Z_n. This would add significant complication to the program, and time to work this out was judged low priority.

LOVSUM.MS

```
> for n from A to B do
> x:=trunc(sqrt(n));
> m:=0:  s:=0:
> a:=n mod 2:
> if a=0
>    then t:=n/2;
> else
>    t:=(n-1)/2:
> fi:
> mm:=4:  ss:=4:
> for j from 2 to t do
>    for k from 1 to t do
>         z:=evalf(2*(cos(2*Pi*k*1/n)+cos(2*Pi*k*j/n))):
>         if abs(z)<4 then m:=max(m,z);
>            else m:=m;
>         fi:
>         if abs(z)<4 then s:=max(s,abs(z));
>            else s:=s;
>         fi:
>    od:
>    mm:=min(mm,m);
>    ss:=min(ss,s);
>    m:=0:  s:=0:
> od:
> lprint(`n=`,n);
> lprint(`best lambda 1 for gen set S={1,__} is`,
>                                   evalf(4-mm));
> lprint(`lowest mu is`,ss);
> od:
```

Selected results of running LOVSUM.MS for n from 10 through 200 are presented below in Table B.2.1.

TABLE B.2.1: SELECTED RESULTS OF LOVSUM FOR n=10 TO 200 WITH BEST RESULTS FOR GENERATING SETS OF THE FORM S={±1,±y}

n	λ_1	μ	n	λ_1	μ	n	λ_1	μ
10	3	1	31	1.10	3.43	107	.366	3.81
11	2.60	2.51	32	1.17	2.83	118	.336	3.66
12	2.27	1.73	33	1.16	3.41	121	.317	3.84
13	2.62	2.65	34	1.14	2.91	125	.310	3.84
14	2.31	2.25	35	1.08	3.47	133	.319	3.84
15	2.38	2.96	40	.922	3.08	142	.281	3.72
16	2	2.62	45	.936	3.57	144	.298	3.70
17	1.95	2.91	50	.788	3.24	152	.268	3.73
18	2	2	55	.658	3.66	160	.269	3.73
19	1.62	3.11	60	.673	3.39	169	.232	3.88
20	1.76	2.62	65	.588	3.70	178	.228	3.78
21	1.51	3.25	70	.526	3.47	184	.235	3.80
22	1.49	2.51	75	.570	3.71	191	.209	3.89
23	1.67	3.13	80	.485	3.51	196	.212	3.79
24	1.55	2.45	85	.475	3.76	199	.200	3.90
25	1.47	3.24	90	.426	3.57	200	.203	3.80
26	1.37	2.65	95	.426	3.78			
27	1.27	3.32	100	.398	3.60			
28	1.33	2.82						
29	1.25	3.35						
30	1.04	2.96						

Using an adapted version of LOVSUM.MS to evaluate λ_1 for various generating sets of size 8 for the Cayley graph based on Z_{1024}, we show the following results:

TABLE B.2.2: VALUES OF λ_1 FOR SELECTED GENERATING SETS OF SIZE 8 ON Z_{1024}

S	λ_1		S	λ_1
{1, 6, 36, 216}	1.201		{1, 7, 32, 224}	1.172
{1, 6, 32, 196}	0.996		{1, 7, 40, 280}	1.172
{1, 6, 35, 210}	1.107		{1, 7, 42, 294}	1.016
{1, 6, 36, 256}	0.754		{1, 7, 42, 273}	0.981
{1, 6, 43, 256}	0.755		{1, 7, 42, 256}	0.967
{1, 6, 42, 273}	1.167		{1, 7, 42, 252}	0.948
{1, 6, 42, 252}	1.004		{1, 7, 38, 266}	0.899
{1, 6, 42, 256}	0.763		{1, 7, 35, 230}	0.797
{1, 6, 37, 222}	0.986		{1, 7, 49, 343}	0.778
{1, 5, 25, 125}	0.584			
{1, 4, 16, 64}	0.163			
{1, 8, 64, 256}	0.586			

The thrust of these results is that there does not seem to be a predictable pattern in producing the best value of λ_1. It seems clear that there is a local maximum at or near the generating set which consists of powers of 6, but it is also true that powers of 7 and some cross-pollination of 6's and 7's show good results. Also, small perturbations in values of the generating elements can have a significant effect on λ_1. Hence, it would seem that examination by exhaustion of candidate generating sets would be the only way to maximize λ_1 for generating sets of any fixed size.

APPENDIX C

COUNTING WALKS IN THE k-REGULAR TREE

FOR

PROOF OF THEOREM 4.4.1

In the proof of Theorem 4.4.1, the claim is made that

$$\rho'(2t) = (1/t)C(2t-2,t-1)k(k-1)^{t-1}$$

where $\rho'(2t)$ is the number of walks of length 2t in the infinite k-regular tree T^k beginning at a vertex v and returning to the same vertex for the first time. Such a walk of length 2t must make t steps away (or "up" a level) from v and t steps back (or "down" a level) toward v.

Consider v to be at zero (or "ground") level. Then we may view a vertex 1 step up (away) from v to be at level 1, a vertex 2 steps up from v to be at level 2, and so on. To move up a level, there are k choices when moving from v at level zero to a vertex at level 1, and k-1 choices for every jump up between higher levels. Moving down a level allows no choice: there is only one path heading back toward v.

Hence, a walk of length 2t which begins at v and returns to v for the first time only at the 2t^th step will have had k choices for its first step, and k-1 choices for each of the t-1 up steps taken. That is, there are $k(k-1)^{t-1}$ distinct walks possible for each allowed distinct string containing t ups and t downs. Then

$$\rho''(2t) = a_t k(k-1)^{t-1}$$

where a_t is the number of distinct allowed strings of length 2t containing t ups and t down steps. Thus, our task reduces to finding a_t.

Claim: $a_{n+1} = a_n a_1 + a_{n-1}a_2 + \ldots + a_2 a_{n-1} + a_1 a_n$

where $a_1=1$ and $a_2=1$.

Clearly, every allowed string must begin with an up and end with a down in order to satisfy the constraints of beginning and ending at vertex v on level zero. Thus, $a_1=1$, since there is only one possible string of length 2 beginning with an up and ending with a down. Similarly, after the initial up, the next entry in the string must also be an up for any allowed string of length greater than or equal to 4, since the walk may not return to vertex v before the last step. Hence, $a_2=1$, with the only allowed such string being up-up-down-down.

To demonstrate the recursion relation claimed, we must find a convenient way to count distinct subsets of allowed strings. As discussed above, the first entry in any such string must be up, while the last must be down. Then any allowed string counted by a_{n+1} will be of the form

up--(string of length 2n)--down.

Observe that the interior string of length 2n begins and ends at level 1, and is restricted to remaining at or above level 1 during its entire run. This allows us to partition our set of allowed strings of length 2(n+1) into subsets defined by the location of the first return to level 1 within the interior string of length 2n. For example, the subset whose interior strings begin with up-down will certainly be distinct from the subset whose interior strings begin with up-up-down-down.

These subsets may be labeled by the length of the fragment of the interior string from its starting point at level 1 to where it first returns to level 1. Since it will require an even number of entries in the interior string to return to level 1, the labels will be the values 2n, 2n-2, 2n-4, and so on down to the value 2. The value 2n corresponds to never returning to level 1 during the entire internal length of the interior string. That is, this subset contains the number of strings containing n ups and n downs which never returns to its starting level until the last step. Such a set is, by definition, counted by a_n. Since $a_1=1$, we have that the size of the subset corresponding to the value 2n is $a_n = a_n \cdot 1 = a_n a_1$.

For the subset corresponding to the value 2n-2, we count strings of length 2n-2 which never return to their original level until the last step (there are a_{n-1} such strings, by definition) and multiply by the number of ways to finish the last 2 steps without going below level 1. But this last quantity is the same as counting the number of ways to do 4 steps without returning to the original level (consider start and finish to be at level zero) until the end, which, by definition, is simply $a_2 = \rho'(2\cdot 2) = \rho'(4)$.

Then the subset labeled with the value 2n-4 corresponds to the product of a_{n-2} and a_3, and so on, until we reach the subset labeled 2, which will correspond to the product of a_1 and a_n. Hence, we have that

$$a_{n+1} = a_n a_1 + a_{n-1} a_2 + \ldots + a_2 a_{n-1} + a_1 a_n$$

as claimed. This recursion relation is apparently a common one for problems in combinatorics, and the sequence of numbers produced by its evaluation are called the Catalan numbers. The closed formula for the value of the Catalan numbers is

$$a_n = (1/n)C(2n-2,n-1)$$

for integer $n \geq 1$, where $C(0,0)=1$ by definition. The development of this closed form may be found in most combinatorics texts. In particular, Jackson and Thoro [JT] demonstrate its validity through a combinatorial problem (Example 7, Chapter 8.1), while Roberts [Rob] evaluates it explicitly through techniques of expansion.

With a_t evaluated, then, we have finally

$$\begin{aligned} \rho'(2t) &= a_t k(k-1)^{t-1} \\ &= (1/t)C(2t-2,t-1)k(k-1)^{t-1} \end{aligned}$$

as desired for Theorem 4.4.1.

APPENDIX D

THE LOVASZ ALGORITHM

A common need for mathematical study of certain problems is to find the eigenvalues of related matrices. Alon and Milman refer to Lovasz [Lov] for a straightforward formula for finding the spectrum of the Cayley graph's adjacency matrix when the underlying group is abelian, and hence all representations are 1-dimensional into the complex numbers.

We present here the essence of Lovasz' argument for evaluation of the spectrum of the adjacency matrix of a Cayley graph, and show its reduction in the abelian case.

Spectrum Evaluation ([Lov]): Let $G=(V,E)$ be the k-regular Cayley graph on $\Gamma=\{x_1,\ldots x_n\}$ with generating set $S=S^{-1}$ ($|S|=k$), and let $A(G)$ be its standard adjacency matrix. Let $T=\{\sigma_1,\ldots,\sigma_n\}$ be the transitive automorphism group on G of order n (described in Theorem 1.2.3) defined by

$$\sigma_i(x) = x \ast x_i \text{ for every } x\in\Gamma.$$

Let π be the regular representation of T over the field of complex numbers. That is, $\pi(\sigma)$ is an nxn permutation matrix which corresponds to the way σ permutes the elements of T. (See Chapter 3.3 and [Dia] for more discussion on the regular representation and representations in general.) Hence, $\pi(\sigma)$ preserves the adjacency of vertices in G, as does its inverse $\pi(\sigma)^{-1}$. Then $\pi(\sigma)$ may be viewed as a change of basis matrix which preserves that adjacency, and so

$$\pi(\sigma)^{-1}A(G)\pi(\sigma) = A(G)$$

or, equivalently,

$$A(G)\pi(\sigma) = \pi(\sigma)A(G).$$

Suppose v were an eigenvector of A(G), with eigenvalue ξ. Then

$$\begin{aligned} A(G)\{\pi(\sigma)v\} = \{A(G)\pi(\sigma)\}v &= \{\pi(\sigma)A(G)\}v \\ &= \pi(\sigma)\{A(G)v\} \\ &= \pi(\sigma)\{\xi v\} \\ &= \xi\{\pi(\sigma)v\}, \end{aligned}$$

which says that $\pi(\sigma)v$ is also an eigenvector of A(G) with eigenvalue ξ.

Let M_ξ be the eigensubspace of A(G) belonging to ξ. Then, by the above argument, M_ξ is invariant under π. That is, π_ξ is a subrepresentation of π on the subspace M_ξ.

Let $\chi_1,\ldots,\chi_t$ be the irreducible characters of T. Refine each subspace M_ξ into its irreducible π-invariant subspaces, which we shall denote N_{ij}, where i runs from 1 to t, j runs from 1 to n_i (the number of isomorphic copies of N_i in the decomposition), and χ_i is the character belonging to the irreducible subrepresentation on N_{ij}. Note that each element of the subspace N_{ij} is an eigenvector of A(G) with the same eigenvalue ξ_{ij}.

Let $b_{ij1},\ldots,b_{ijn_i}$ be an orthogonal basis for the subspace N_{ij}. (The b's are indexed by n_i, which is both the dimension of each N_{ij} and the number of isomorphic copies of N_i in the decomposition.) Then, since the character of any subrepresentation is the trace of the block submatrix of the representation, and since the trace is independent of a change of basis (for diagonalization), we have that

$$\chi_i(\sigma) = tr(\Pi_{ij}(\sigma)) = \sum_{v=1}^{n_i} b_{ijv}\Pi(\sigma)b_{ijv}.$$

Then we may substitute for $\chi_i(\sigma)$ in the following

$$\sum_{i=1}^{t}\sum_{j=1}^{n_i} \xi_{ij}{}^q\chi_i(\sigma) = \sum_{i=1}^{t}\sum_{j=1}^{n_i} \xi_{ij}{}^q\sum_{v=1}^{n_i} b_{ijv}\Pi(\sigma)b_{ijv}$$

$$= \sum_{i=1}^{t}\sum_{j=1}^{n_i}\sum_{v=1}^{n_i} b_{ijv}\Pi(\sigma)\,(\xi_{ij}{}^q b_{ijv})$$

$$= \sum_{i=1}^{t}\sum_{j=1}^{n_i}\sum_{v=1}^{n_i} b_{ijv}\Pi(\sigma)\,(A^q b_{ijv})$$

since $\xi_{ij}{}^q$ is an eigenvalue of A^q for any element in any of the N_{ij}'s, as noted above. But this is just the sum of the diagonals (after diagonalization) of all the block submatrices of $\pi(\sigma)A^q$ corresponding to the eigenspaces of A. Noting that the traces of similar matrices are equal (proved during Theorem 4.4.1), we see that the right-hand side is thus equivalent to the trace of $\pi(\sigma)A^q$.

Let $p_{\sigma,q}$ be the number of walks of length q from any vertex in G to its image under the automorphism σ. That is, it is the sum over all i of the walks of length q from vertex x_i to vertex $\sigma(x_i)$.

Suppose $\sigma_i(\sigma_j)=\sigma_k$. By the definition of the regular representation, this means that row k of the permutation matrix σ_i has a 1 in column j. Also, by the definition of the automorphism group T, we have

$$\begin{aligned} \sigma_k(x) = [\sigma_i(\sigma_j)](x) &= \sigma_i(x{*}x_j) \\ &= (x{*}x_j){*}x_i \\ &= x{*}(x_j{*}x_i) \end{aligned}$$

and hence $x_k = x_j{*}x_i$. That is to say, $\sigma_i(x_j) \mapsto x_k$.

By the nature of matrix algebra, left-multiplication by

$\pi(\sigma_i)$ exchanges the rows of $A^q(G)$. If row k of $\pi(\sigma_i)$ has a 1 in column j, then the jth row of $A^q(G)$ becomes row k of $\pi(\sigma_i)A^q(G)$. Then $[\pi(\sigma_i)A^q(G)]_{kk} = [A^q(G)]_{jk}$. Recalling that a property of adjacency matrices is that the qth power of such a matrix has as its jkth entry the number of walks of length q between the vertices x_j and x_k, we see that $[A^q(G)]_{jk}$ is simply the number of walks of length q from x_j to x_k. But x_k is the image of x_j via σ_i. That is, $[A^q(G)]_{jk}$ is the number of walks of length q from x_j to $x_k=\sigma_i(x_j)$. Then $tr(\pi(\sigma_i)A^q(G))$, found by summing over the entire diagonal of $\pi(\sigma_i)A^q(G)$, is the sum of all such $[A^q(G)]_{jk}$'s, which is just $p_{\sigma,q}$, by definition. Thus, we have that

$$\sum_{i=1}^{t}\sum_{j=1}^{n_i} \xi_{ij}{}^q\chi_i(\sigma) = tr(\pi(\sigma_i)A^q(G)) = p_{\sigma,q}.$$

Since ξ_{ij} and $p_{\sigma,q}$ are real numbers, we know also that

$$\sum_{i=1}^{t}\sum_{j=1}^{n_i} \xi_{ij}{}^q\overline{\chi_i(\sigma)} = p_{\sigma,q},$$

where $\overline{\chi_i(\sigma)}$ is the complex conjugate of $\chi_i(\sigma)$.

Then, for a fixed i, we have from substitution and a change of indices that

$$\sum_{\sigma\in T} p_{\sigma,q}\chi_i(\sigma) = \sum_{\sigma\in T}\sum_{\mu=1}^{t}\sum_{j=1}^{n_\mu} \xi_{\mu j}{}^{q}\overline{\chi_\mu(\sigma)}\chi_i(\sigma)$$

$$= \sum_{\mu=1}^{t}\sum_{j=1}^{n_\mu} \xi_{\mu j}{}^{q}\sum_{\sigma\in T} \overline{\chi_\mu(\sigma)}\chi_i(\sigma) .$$

Recalling (from [Dia], Chapter 2, Theorem 3) that the characters of irreducible representation are orthonormal(and that complex numbers are defined to be orthogonal when the product of their conjugates are zero), we have that $\overline{\chi_\mu(\sigma)}\chi_i(\sigma) = 0$ for any $\mu\neq i$, while $\overline{\chi_i(\sigma)}\chi_i(\sigma) = 1$. Hence, the only nonzero terms in the right-hand side occur when $\mu=i$, and we have

$$\sum_{\mu=1}^{t}\sum_{j=1}^{n_\mu} \xi_{\mu j}{}^{q}\sum_{\sigma\in T} \overline{\chi_\mu(\sigma)}\chi_i(\sigma) = \sum_{j=1}^{n_i} \xi_{ij}{}^{q}\sum_{\sigma\in T} \overline{\chi_i(\sigma)}\chi_i(\sigma)$$

$$= n\sum_{j=1}^{n_i} \xi_{ij}{}^{q}.$$

Thus,

$$\sum_{\sigma\in T} p_{\sigma,q}\chi_i(\sigma) = n\sum_{j=1}^{n_i} \xi_{ij}{}^{q}.$$

In general, solving such equations is a procedure involving finding determinants of possibly large matrices. (They will have dimension less than n, but still may be difficult to evaluate.) However, we can make two reductions that make the solution much simpler to find. First, if we are concerned only with finding the eigenvalues of A (and not the higher powers) then we may set the path length q to 1, yielding

$$\sum_{\sigma\in T} p_{\sigma 1}\chi_i(\sigma) = n\sum_{j=1}^{n_i} \xi_{ij}.$$

A path of length one implies adjacency between a vertex and its image under $\pi(\sigma)$, which occurs in the Cayley graph of G only when σ corresponds to an element of Γ that is in the generating set S, which we shall denote with the somewhat loose notation of $\sigma\in S$. Hence, when "$\sigma\in S$", we have that $p_{\sigma 1} = p_\sigma = n$, since there are n such adjacencies, and when "$\sigma\notin S$", $p_\sigma=0$. The left-hand side thus simplifies as

$$\sum_{\sigma\in T} p_{\sigma 1}\chi_i(\sigma) = n\sum_{\sigma\in S} \chi_i(\sigma) .$$

Second, if we consider only the abelian case for Γ, then T is abelian and n_i=1 for every i. Thus

$$n\sum_{j=1}^{n_i} \xi_{ij} = n\xi_i,$$

and so

$$n\sum_{\sigma\in S} \chi_i(\sigma) = n\xi_i,$$

or

$$\sum_{\sigma\in S} \chi_i(\sigma) = \xi_i. \qquad (*)$$

Since n_i=1, the dimension of each irreducible representations is one, and so the trace (character) of each representation and the representation itself are the same. For a cyclic group Z_n, we showed in Chapter 3.5 that the irreducible unitary representations π_i are the n distinct n^{th} roots of unity. Then, for $\Gamma = Z_n = \{0,1,2,\ldots,n-1\}$, we have the cyclic group of automorphisms $T=\{\sigma_0,\ldots,\sigma_{n-1}\}$ where σ_j is the permutation on Γ defined by

$$\sigma_j(i) = i+j(\text{mod } n) \text{ for every } i\in\Gamma.$$

Hence, $\sigma_j \circ \sigma_k = \sigma_{j+k} = \sigma_{k+j}$, and the relations in T tracked by the subscripts of its elements are identical to the relations of those subscript values as elements of Γ. That is, T and Γ are isomorphic by the mapping of subscript to corresponding element, and so we may consider the representation on T and a representation on Γ to be the same. Thus, we may define the representations by

$$\pi_t(\sigma_j) = \pi_t(j) = \cos(2\pi tj/n) + \imath\sin(2\pi tj/n)$$

for every t from 0 through n-1.

Then equation (*) above may be written as

$$\xi_t = \sum_{s \in S} [\cos(2\pi ts/n) + \imath\sin(2\pi ts/n)].$$

Since S is closed under inverses and the identity is not in S (both by definition of Cayley graphs) we will always have one of two cases for evaluating these sums:

i) s_i is its own inverse. Then $s_i=n/2$, and

$\pi_t(s_i) = \cos(2\pi t/2) + \iota\sin(2\pi t/2) = \pm1 + 0\iota = \pm1$,

depending on whether t is even or odd.

ii) s_i is not its own inverse. Then by definition S contains the inverse $s_i^{-1}=n-s_i$ the sum of these pairs is

$$\begin{aligned}\pi_t(s_i) + \pi_t(s_i^{-1}) &= \cos(2\pi s_i t/n) + \iota\sin(2\pi s_i t/n)\\ &\quad + \cos(2\pi s_i^{-1}t/n) + \iota\sin(2\pi s_i^{-1}t/n)\\ &= \cos(2\pi s_i t/n) + \iota\sin(2\pi s_i t/n)\\ &\quad + \cos(2\pi(n-s_i)t/n) + \iota\sin(2\pi(n-s_i)t/n)\\ &= \cos(2\pi s_i t/n) + \iota\sin(2\pi s_i t/n)\\ &\quad + \cos(-2\pi s_i t/n) + \iota\sin(-2\pi s_i t/n)\\ &= \cos(2\pi s_i t/n) + \iota\sin(2\pi s_i t/n)\\ &\quad + \cos(2\pi s_i t/n) - \iota\sin(2\pi s_i t/n)\\ &= 2\cos(2\pi s_i t/n).\end{aligned}$$

Then in the first case, n is even, k is odd, and there is exactly one element that is its own inverse ($s_{(k+1)/2} = n/2$), and we have

$$\xi_t = 2\cos(2\pi s_1 t/n) +\ldots+ 2\cos(2\pi s_{(k-1)/2} t/n) + \cos(2\pi t/2)$$

while in the second case k is even and each element may be paired with its inverse, yielding

$$\xi_t = 2\cos(2\pi s_1 t/n) + \ldots + 2\cos(2\pi s_{k/2} t/n).$$

Note that, since $\cos(2\pi s_i t/n) = \cos(2\pi s_i (n-t)/n)$ regardless of which i is used, to find all possible eigenvalues we need only run through the values of t from 0 through n/2 (if n is even) or (n+1)/2 (if n is odd).

For our purposes, there are two resulting values of interest from these calculations. The first is the spectral radius μ which may be found from

$$\mu = max\{|\xi_t|: |\xi_t|<k\}.$$

The second is λ_1, the smallest nonzero eigenvalue of $Q=K-A(G)$, which will be found by

$$\lambda_1 = k - \mathit{max}\{\xi_t\colon\ \xi_t<m\}.$$

Note that $\mathit{max}\{\xi_t\colon\ \xi_t<k\} \le \mathit{max}\{|\xi_t|\colon\ |\xi_t|<k\}$, so that we have $\lambda_1 \ge k-\mu$ (or, equivalently, $\mu \ge k-\lambda_1$) in every case. It is a rather straightforward task, then, to evaluate μ and λ_1 for the case of a group of the form Z_n and a specified generating set S; simply running through all necessary values of t and looking for the appropriate maximums yields our desired constants. The Maple program in Appendix B is designed to do this. The following theorem evaluates for a specific case used quite often in this thesis.

THEOREM D.1: Let G=G(V,E) be the Cayley graph on Z_n with generating set $S' = \{1,x,1^{-1},x^{-1}\}$, where x is the truncated square root of n. Then we have that

$$\lambda_1 = \begin{cases} 2-2\cos(2\pi/n^{1/2}) & \text{if n is a perfect square;} \\ 4-2[\cos(2\pi/n)+\cos(2\pi x/n)] & \text{otherwise.} \end{cases}$$

***Proof*:** Consider all the possible values of t between 1 and n/2, ignoring the trivial 0 which gives $\xi_0=4$:

For t=1: $\xi_1 = 2\cos(2\pi/n) + 2\cos(2\pi x/n)$.

For $1 < t < x$: $\xi_t = 2\cos(2\pi t/n) + 2\cos(2\pi tx/n) \leq \xi_1$

since $\cos(2\pi/n) \geq \cos(2\pi t/n)$ and $(t+1)x \leq x^2 \leq n$ implying that $2\pi tx/n$ is at least $2\pi x/n$ away from 0 on the unit circle and hence $\cos(2\pi x/n) \geq \cos(2\pi tx/n)$.

For t=x: $\xi_x = 2\cos(2\pi x/n) + 2\cos(2\pi x^2/n)$.

i) If $x^2=n$, then $\cos(2\pi x^2/n)=\cos(2\pi n/n)=\cos(2\pi)=1$, so $\xi_x > \xi_1$.

ii) If $x^2<n$, then $\cos(2\pi x^2/n) \leq \cos(2\pi/n)$ since x^2/n is at least $2\pi/n$ away from 0 on the unit circle.

Hence, we get that $\xi_x \leq \xi_1$.

$n/2 \geq t > x$: $\xi_t = 2\cos(2\pi t/n) + 2\cos(2\pi xt/n)$.

Observe that the first term of this sum will range between $2\cos(2\pi(x+1)/n)$ and $2\cos(\pi)=-2$. Hence, its largest possible contribution to this sum is when $t=x+1$. Note that the second term could possibly contribute $+2$ to the sum if $xt=n$. This is larger than the contribution of the first term for the case of $t=1$. However, since the arc traced between the angles 0 rads and $2\pi/n$ rads is nearly vertical on the unit circle, not as much is contributed to the sum by this change as is lost by the difference in the cosines from the more horizontal arc traced between $2\pi x/n$ rads and $2\pi(x+1)/n$ rads. That is

$$\cos(2\pi)-\cos(2\pi/n) \le \cos(2\pi x/n)-\cos(2\pi(x+1)/n).$$

Hence, if n is not a perfect square, we have shown that the largest ξ_t possible occurs when $t=1$, and then

$$\lambda_1 = 4 - \xi_1$$

$$= 4 - 2\cos(2\pi/n) - 2\cos(2\pi x/n)$$

$$= 4 - 2[\cos(2\pi/n) + \cos(2\pi x/n)].$$

If $n=x^2$ is a perfect square, the largest ξ_t possible occurs when $t=x$, and then

$$\lambda_1 = 4 - \xi_x$$

$$= 4 - 2\cos(2\pi/n^{½}) - 2\cos(2\pi)$$

$$= 2 - 2\cos(2\pi/n^{½}). \qquad \square$$

Conceivably, similar reasoning can be carried out for numerous other cases, but it seems much more efficient to simply allow the computer program in Appendix B to generate the results as desired.

APPENDIX E

PATTERN CHARTS FOR ALGORITHMIC DIAMETER AND AVERAGE DIAMETER OF Z_n-BASED CAYLEY GRAPHS

The following tables are designed to show the number of steps required to reach a given vertex from the zero vertex for the designated generating set. This aids in establishing the algorithmic average and full diameters as explored in Chapter 6. Clear patterns emerge, so formulas could be established, but the visual comparisons are useful in providing immediate feedback as to best choices for a specific situation. Note that all entries are based on using the greedy algorithm described in Chapter 6, unless otherwise noted.

TABLE E.1: DISTANCE PATTERNS FOR VARIOUS GENERATING SETS

distance from 0		Number of Steps Required to Reach That Distance Using the Listed Generating Set. (All values are using the greedy algorithm, except where noted with *.)							
		1,4	1,5	1,6	1,7	1,8		1,4,16	1,7*
0		0	0	0	0	0		0	0
1		1	1	1	1	1		1	1
2		2	2	2	2	2		2	2
3		2	3	3	3	3		2	3
4		1	2	3	4	4		1	4
5		2	1	2	3	4		2	3
6		3	2	1	2	3		3	2
7		3	3	2	1	2		3	1
8		2	4	3	2	1		2	2
9		3	3	4	3	2		3	3
10		4	2	4	4	3		4	4
11		4	3	3	5	4		3	3*
12		3	4	2	4	5		2	4
13		4	5	3	3	5		3	3
14		5	4	4	2	4		3	2
15		5	3	5	3	3		2	3
16		4	4	5	4	2		1	4
17		5	5	4	5	3			
18		6	6	3	6	4			
19		6	5	4	5	5			
20		5	4	5	4	6			
	Σ	70	66	63	66	66		37	44

Note that {1,7} using a backdoor route on Z_{32} reaches vertex 11 in 3 steps, instead of 5.

TABLE E.1 CONTINUED

dist	1, 8	1, 10	1, 12
0	0	0	0
1	1	1	1
2	2	2	2
3	3	3	3
4	4	4	4
5	4	5	5
6	3	5	6
7	2	4	6
8	1	3	5
9	2	2	4
10	3	1	3
11	4	2	2
12	5	3	1
13	5	4	2
14	4	5	3
15	3	6	4
16	2	6	5
17	3	5	6
18	4	4	7
19	5	3	7
20	6	2	6
Σ	66	70	82

dist	1,8	1, 10	1, 12
21	6	3	5
22	5	4	4
23	4	5	3
24	3	6	2
25	4	7	3
26	5	7	4
27	6	6	5
28	7	5	6
29	7	4	7
30	6	3	8
31	5	4	8
32	4	5	7
33		6	6
34		7	5
35		8	4
36		8	3
37		7	4
38		6	5
39		5	6
40		4	7
Σ	62	110	102

dist	1, 10	1, 12
41	5	8
42	6	9
43	7	9
44	8	8
45	9	7
46	9	6
47	8	5
48	7	4
49	6	5
50	5	6
51		7
52		8
53		9
54		10
55		10
56		9
57		8
58		7
59		6
60		5
61		6
62		7
63		8
64		9
Σ	70	176

TABLE E.2: PATTERNS FOR MULTIPLES FOUND IN TABLE 6.6.1

This chart provides the number of steps to reach the "landing pad" vertices for two cases in Table 6.6.1. The left-hand column of each pair of columns shows the distance from zero while the right-hand column of each pair gives the number of steps required by the greedy algorithm to reach that "landing pad" vertex with the given generating elements.

Distance	32, 224		Distance	36, 216
0	0		0	0
32	1		36	1
64	2		72	2
96	3		108	3
128	4		144	3
160	3		180	2
192	2		216	1
224	1		252	2
256	2		288	3
288	3		324	4
320	4		360	4
352	5		396	3
384	4		432	2
416	3		468	3
448	2		504	4
480	3			
512	4			

APPENDIX F

NETWORK DESIGN CORRECTION

FOR

PROOF OF THEOREM 4.3.5

In the proof of Theorem 4.3.5, a network N(G) is constructed on an underlying graph G=G(V,E), where G is an (n,d,c)-magnifier, as follows: N(G) has the vertex set $\{s,t\}\cup X\cup Y$, where s is the source, t is the sink, and $X=V^{+}$ and Y=V are disjoint sets of vertices. (The vertex set $V^{+}\subset V$ has been selected to contain no more than half of the vertices in G.) Construct the arcs of N(G) as follows:

(a) For every $u\in X$, the arc (s,u) is assigned capacity 1+c.

(b) For every $u\in X$ and $v\in Y$, the arc (u,v) is assigned capacity 1 if $(u,v)\in E$ or u=v; capacity 0 otherwise.

(c) For every $v \in Y$, the arc (v,t) is assigned capacity 1.

As noted in the text of Chapter 4, part (b) of these parameters for construction differs from that laid out by Alon [A], in that he omits explicitly assigning a capacity of 1 to each arc joining a vertex $u \in X=V^+$ with its subscript partner $v \in Y$. (Here, subscript partner is used in the sense that $u \in X$ corresponds to the same vertex in the underlying graph G as does $v \in Y$. Thus, if each $v \in X$ is assigned the subscript of the vertex it corresponds to in G, and the same format is followed for the subscripts of each $u \in Y$, then subscript partners have the same subscript.)

Alon later claims that, for any subset $U \subset X$, the set of vertices which are "neighbors" of the vertices in U (i.e., vertices in Y with at least one arc of capacity 1 from a vertex in U), denoted by $N(U)$, satisfies by the magnifying properties of the underlying graph that

$$|N(U)| \geq (1+c)|U|.$$

However, the definition of an (n,d,c)-magnifier simply provides that the **new** neighbors of U, denoted by $|N(U)-U|$, satisfies the statement

$$|N(U)-U| \geq c \cdot |U|.$$

But N(U) is simply the subset of Y corresponding to vertices in the underlying graph G which are adjacent to at least one vertex in G corresponding to a vertex in U. This definition does **not** guarantee that every vertex in U is included in N(U), since a vertex in U may not itself be adjacent to any other vertex in U ("adjacent" with regard to the corresponding set of vertices in G). It is **possible** that the construction of the set V^+ as the no-more-than-half of the vertices corresponding to the positive entries in a non-constant eigenvector of Q can lead to this guarantee. Alon makes no mention of this, nor does there appear to be any basis for such a claim. However, it has not been disproved as a possibility.

Assuming that the construction of V^+ is not responsible for this claim, we may only conclude that Alon intended for the capacity of arcs between subscript partners to be assigned a value of 1. Without such an assignment of capacities, we have by basic set theory only that

$$|N(U)-U| + |U| \geq |N(U)|$$

which does not allow the conclusion that $|N(U)| \geq |U| + c\cdot|U|$, as claimed by Alon. This may only be concluded if we know that arcs between subscript partners have capacity 1, so that $|N(U)| + |U| = |N(U)|$.

INDEX