Delft University of Technology

Faculty of Electrical Engineering, Mathematics and Computer Science
Delft Institute of Applied Mathematics

The Neighbour-Sum Problem on Graphs

A thesis submitted to the
Delft Institute of Applied Mathematics
in partial fulfilment of the requirements
for the degree of

Bachelor of Science
in
Applied Mathematics

by

Ruth Cuypers

Delft, The Netherlands
July 2026

Supervisor:
Dr. D.C. Gijswijt

Preface

In front of you is my Bachelor's thesis, written as the final part of my Bachelor's programme in Applied Mathematics at Delft University of Technology.

Throughout my studies I have particularly enjoyed graph theory, linear algebra, and solving mathematical puzzles. When I was choosing a topic for my Bachelor's thesis, the neighbour-sum problem immediately caught my attention. I liked how a seemingly simple puzzle naturally leads to questions in graph theory, linear algebra, and spectral analysis. This made the project a perfect match for my mathematical interests.

Working on this thesis has been both challenging and rewarding. Besides learning many new mathematical techniques, I especially enjoyed searching for patterns, proving theoretical results, and complementing these with computational experiments. It was fascinating to see how different areas of mathematics come together in the study of a single problem.

I would like to thank my supervisor, Dion Gijswijt, for his guidance and support throughout this project. I especially appreciated how approachable he was and how easy it was to communicate and discuss new ideas. Our meetings were always enjoyable and often led to interesting new directions for the research.

Contents

1

Layman's summary

Many real-world systems can be viewed as networks, such as social networks, road networks, or communication networks. In these networks, the vertices represent objects, such as people or cities, while the edges represent the connections between them. In this thesis, we study a mathematical puzzle on such networks. The goal is to assign a number to each vertex in such a way that the number at every vertex is equal to the sum of the numbers assigned to its neighbouring vertices. While assigning the value zero everywhere always works, the interesting question is whether other solutions exist.

Although this puzzle is easy to state, answering this question requires tools from several areas of mathematics. By translating the problem into linear algebra, we can determine whether a network has a non-trivial solution by studying a matrix associated with the graph.

For highly symmetric networks, such as paths, cycles, and circulant graphs, complete mathematical conditions can be derived that determine exactly when a solution exists. For more general networks, especially trees, this becomes much more difficult. Therefore, in addition to theoretical results, computer experiments were performed on thousands of different trees to investigate which structural properties make a solution more likely.

The experiments show that larger trees are more likely to admit a solution. They also suggest that trees with a moderate amount of branching are more likely to have a solution than either very simple path-like trees or highly branched trees. Together, the theoretical results and computational experiments provide new insight into this mathematical puzzle and illustrate how algebra, graph theory, and computer experiments can be combined to study complex problems.

2

Summary

This thesis investigates the neighbour-sum problem on graphs. Given a graph, the objective is to determine whether there exists a non-trivial assignment of real numbers to its vertices such that the value at every vertex equals the sum of the values assigned to its neighbours. This problem can be reformulated as the eigenvalue problem $Ax = x$, where A denotes the adjacency matrix of the graph. Consequently, a graph admits a non-trivial solution if and only if 1 is an eigenvalue of its adjacency matrix.

The first part of the thesis develops theoretical results for several graph classes. Explicit characterizations are obtained for path graphs and cycle graphs by determining their spectra. These results are extended to Cayley graphs over finite abelian groups using Fourier analysis on finite groups. By expressing the eigenvalues in terms of group characters, explicit conditions are derived for circulant graphs, together with computational experiments illustrating the influence of the generating set.

The second part focuses on trees. Several theoretical observations are established for special families of trees, including stars, double stars, and caterpillar trees. For caterpillar trees, the neighbour-sum problem is reduced to studying the determinant of a tridiagonal matrix. Furthermore, using a result of Schwenk, it is shown that the proportion of solvable trees tends to one as the number of vertices tends to infinity.

Finally, computational experiments were performed on all non-isomorphic trees with up to twenty vertices. The experiments indicate that the proportion of solvable trees increases with the number of vertices. Moreover, trees with a moderate number of leaves, moderate maximum degree, and intermediate diameter appear more likely to admit non-trivial solutions than highly branched trees. The solution space is typically one-dimensional, although rare examples with much higher dimensions were identified. These observations suggest several directions for future research toward a structural characterization of solvable trees.

3

Introduction

The goal of this thesis is to study the so-called *neighbour-sum problem* on graphs. Given a graph, we assign numbers to its vertices such that the value at each vertex equals the sum of the values of its neighbouring vertices. The main question is to determine for which graphs there exists a non-trivial solution, that is, a solution in which not all values are zero.

This problem was recently studied by Dutta, Mandal, Gupta and Chatterjee in the context of chessboards [1]. They proved that an $n \times n$ chessboard admits a non-trivial solution if and only if $6 \mid (n+1)$, and also investigated several generalizations, including rectangular and toroidal boards. Motivated by these results, we study the neighbour-sum problem in a graph-theoretic setting.

For every graph, the neighbour-sum condition can be written in matrix form as $Ax = x$, where A denotes the adjacency matrix of the graph. Consequently, a graph admits a non-trivial solution if and only if 1 is an eigenvalue of A. This reformulation allows the problem to be studied using techniques from linear algebra and spectral graph theory.

For highly symmetric graph classes, such as paths, cycles, and circulant graphs, the spectrum can often be determined explicitly, leading to complete characterizations of solvability. For more general graphs, such as trees, explicit spectral formulas are typically unavailable, making the problem considerably more difficult.

This thesis is organised as follows. Chapter 4 introduces the neighbour-sum problem together with the necessary background on graphs, Cayley graphs, and circulant graphs. In Chapter 5 we completely characterize the existence of non-trivial solutions for path and cycle graphs. Chapter 6 extends these results to Cayley graphs over finite abelian groups and circulant graphs using Fourier analysis on finite groups. Chapter 7 investigates trees theoretically by studying several special families, including star graphs, double stars, and caterpillar trees, and presents an asymptotic solvability result. Finally, Chapter 8 contains computational experiments on non-isomorphic trees with up to twenty vertices, examining how structural properties such as the number of leaves, maximum degree, diameter, and the dimension of the solution space influence the existence of non-trivial solutions.

4

Problem formulation and preliminaries

In this chapter, the neighbour-sum problem is formulated mathematically and the necessary preliminaries are introduced. The definitions of graphs, graph isomorphisms, paths, trees, and adjacency matrices follow the standard terminology of [9]. The definitions of Cayley graphs and circulant graphs are based on [5].

4.1. Problem formulation

The following definitions are standard in graph theory and linear algebra.

Definition 4.1.1. *A **graph** $G = (V, E)$ consists of a finite set of vertices V and a set of edges $E \subseteq V \times V$.*

For a graph G, we denote its vertex set by $V(G)$ and its edge set by $E(G)$.

Definition 4.1.2. *The **adjacency matrix** A of a graph G is defined by*

$$A_{ij} = \begin{cases} 1 & \text{if vertices } i \text{ and } j \text{ are adjacent,} \\ 0 & \text{otherwise.} \end{cases}$$

We now formulate the main problem studied in this thesis.

Problem 4.1.1. *Given a graph $G = (V, E)$, determine whether there exists a non-trivial assignment $x_v \in \mathbb{R}$ for every vertex $v \in V$ such that $x_v = \sum_{u \sim v} x_u$ for all vertices $v \in V$.*

In other words, the problem asks whether it is possible to assign values to the vertices of a graph such that the value at every vertex is exactly equal to the sum of the values assigned to its neighbouring vertices. The trivial assignment, where every vertex receives the value 0, always satisfies this condition. Therefore, throughout this thesis we are interested in the existence of non-trivial assignments.

In matrix form, this condition can be written as $Ax = x$, where A is the adjacency matrix of the graph and x denotes the vector of numbers on the vertices. Thus, the problem reduces to determining when there exists a non-zero vector x satisfying $Ax = x$.

Theorem 4.1.1. *A graph admits a non-trivial solution to the neighbour-sum problem if and only if the adjacency matrix has eigenvalue* 1.

Proof. Rewriting the equation $Ax = x$ gives $(A - I)x = 0$. Therefore, a non-trivial solution exists if and only if there is a non-zero vector x satisfying $(A - I)x = 0$. By definition, this is precisely the condition that 1 is an eigenvalue of the adjacency matrix A. □

Throughout this thesis, we consider finite, undirected graphs without loops or multiple edges. Since the adjacency matrix of an undirected graph is symmetric, all eigenvalues of A are real.

We now consider the following example to illustrate the problem further.

Example 4.1.1. *Consider the tree G in Figure 4.1.*

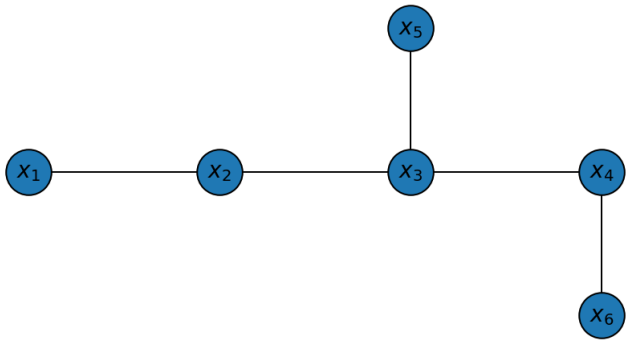

Figure 4.1: Graph G

Then G has adjacency matrix

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}.$$

This is the path P_5 with one extra leaf attached to the middle vertex. Using Python, we compute the eigenvalues of A and find that 1 *is an eigenvalue. Thus by Theorem 4.1.1, G has a non-trivial solution to the neigbour-sum puzzle.*

A corresponding eigenvector is $x = \begin{pmatrix} 1 \\ 1 \\ 0 \\ -1 \\ 0 \\ -1 \end{pmatrix}$ *which is thus a solution that assigns values to each vertex of G.*

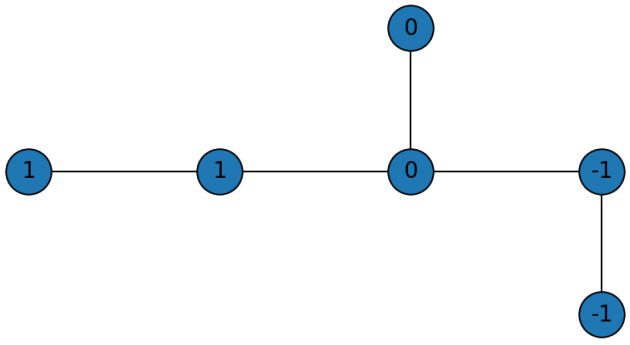

Figure 4.2: Solution for graph G

And Indeed, $Ax = \begin{pmatrix} 1 \\ 1 \\ 0 \\ -1 \\ 0 \\ -1 \end{pmatrix} = x.$

Thus, this tree admits a non-trivial solution to the neighbour-sum problem.

4.2. Cayley graphs

In order to study graphs with a high degree of symmetry, we introduce Cayley graphs. These graphs arise from algebraic structures, namely groups, and provide a setting in which tools from Fourier analysis can be applied, which will be done in Chapter 6.

Definition 4.2.1. *Let G be a finite group and let $S \subseteq G$ be a generating set such that $S = S^{-1}$ and $e \notin S$, where e denotes the identity element of G. The* ***Cayley graph*** $\mathrm{Cay}(G, S)$ *is the graph with vertex set G, where two vertices $g, h \in G$ are adjacent if and only if there exists $s \in S$ such that $h = gs$.*

The condition $S = S^{-1}$ ensures that the graph is undirected. Indeed, if $h = gs$ for some $s \in S$, then $g = hs^{-1}$, and also since $s^{-1} \in S$, adjacency is symmetric. The condition $e \notin S$ prevents loops, since otherwise every vertex g would satisfy $g = ge$ and would therefore be adjacent to itself.

Cayley graphs are particularly useful in this context because of their high degree of symmetry. In particular, when the group G is abelian, the eigenvalues of the adjacency matrix can be computed explicitly using characters, which form the basis of Fourier analysis on finite groups. This makes it possible to analyze the neighbour-sum condition in a direct and explicit way.

4.3. Circulant graphs

A particularly interesting class of Cayley graphs arises when the underlying group is the cyclic group $\mathbb{Z}_n$. In this case, the resulting graphs are called circulant graphs.

Definition 4.3.1. *Let $S \subseteq \mathbb{Z}_n \setminus \{0\}$ be a generating set such that $S = -S$. The* ***circulant graph*** $\mathrm{Cay}(\mathbb{Z}_n, S)$ *has vertex set $V = \{0, 1, \ldots, n-1\}$, where two vertices $i, j \in \mathbb{Z}_n$ are adjacent if and only if $j \equiv i + s \pmod{n}$ for some $s \in S$.*

The condition $S = -S$ ensures that the graph is undirected. Indeed, if $j \equiv i + s \pmod{n}$ for some $s \in S$, then $i \equiv j + (-s) \pmod{n}$, and since $-s \in S$, the adjacency matrix is symmetric.

Circulant graphs are highly symmetric, as their structure is invariant under cyclic shifts of the vertices. This symmetry makes them particularly suitable for spectral analysis. In fact, their eigenvalues can be computed explicitly using Fourier methods, which will allow us to analyze the neighbour-sum condition in a direct way.

The simplest example of a circulant graph is obtained by taking $S = \{1, -1\}$, which yields the cycle graph C_n. This case will be studied in detail in the next chapter.

5

Paths and cycles

In this chapter, we study the neighbour-sum problem for two fundamental classes of graphs: path graphs and cycle graphs. These graphs serve as basic examples that illustrate the main ideas of the problem. In particular, they allow for explicit computations of eigenvalues, which makes it possible to determine when non-trivial solutions exist.

5.1. Path graphs

First we define a path graph, and then we investigate when a non-trivial solution exists for paths.

Definition 5.1.1. *A **path graph** P_n is the graph with vertex set $V = \{1,2,\dots,n\}$, where vertices i and j are adjacent if and only if $|i-j| = 1$. Thus, each vertex is connected to its immediate neighbours along a line.*

The adjacency matrix A of P_n is given by $A = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 1 & 0 & 1 & \cdots & 0 \\ 0 & 1 & 0 & \ddots & 0 \\ \vdots & \vdots & \ddots & \ddots & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$.

This is a tridiagonal matrix, since the only non-zero entries are on the subdiagonal and superdiagonal.

Proposition 5.1.1. *The eigenvalues of the adjacency matrix of the path graph P_n are*

$$\lambda_k = 2\cos\left(\frac{k\pi}{n+1}\right), \qquad k = 1,2,\dots,n.$$

Proof. We follow the proof of Martin-Hernandez et al. [2]. Let A be the adjacency matrix of P_n. We show that, for each $k = 1,\dots,n$, the vector $v^{(k)} \in \mathbb{R}^n$ with entries

$$v_j^{(k)} = \sin\left(\frac{jk\pi}{n+1}\right), \qquad j = 1,\dots,n,$$

is an eigenvector of A.

For an interior vertex $j = 2,\dots,n-1$, we have $(Av^{(k)})_j = v_{j-1}^{(k)} + v_{j+1}^{(k)}$.

Using the identity $\sin(a-b) + \sin(a+b) = 2\sin(a)\cos(b)$, with $a = \frac{jk\pi}{n+1}$ and $b = \frac{k\pi}{n+1}$, we obtain

$$v_{j-1}^{(k)} + v_{j+1}^{(k)} = 2\cos\left(\frac{k\pi}{n+1}\right) v_j^{(k)}.$$

Thus,

$$(Av^{(k)})_j = 2\cos\left(\frac{k\pi}{n+1}\right) v_j^{(k)}.$$

It remains to verify the formula at the endpoints.

For $j = 1$, the vertex 1 is adjacent only to vertex 2, so $(Av^{(k)})_1 = v_2^{(k)} = \sin\left(\frac{2k\pi}{n+1}\right)$
Using the identity $\sin(a-b)+\sin(a+b) = 2\sin(a)\cos(b)$ with $a = \frac{k\pi}{n+1}$ and $b = \frac{k\pi}{n+1}$, we obtain

$$(Av^{(k)})_1 = 2\cos\left(\frac{k\pi}{n+1}\right)\sin\left(\frac{k\pi}{n+1}\right) = 2\cos\left(\frac{k\pi}{n+1}\right)v_1^{(k)}.$$

For $j = n$, we have $(Av^{(k)})_n = v_{n-1}^{(k)} = \sin\left(\frac{(n-1)k\pi}{n+1}\right)$.
Applying the identity $\sin(a-b)+\sin(a+b) = 2\sin(a)\cos(b)$ with $a = \frac{nk\pi}{n+1}$ and $b = \frac{k\pi}{n+1}$, and using $\sin(k\pi) = 0$, yields

$$(Av^{(k)})_n = 2\cos\left(\frac{k\pi}{n+1}\right)\sin\left(\frac{nk\pi}{n+1}\right) = 2\cos\left(\frac{k\pi}{n+1}\right)v_n^{(k)}.$$

Therefore, $(Av^{(k)})_j = 2\cos\left(\frac{k\pi}{n+1}\right)v_j^{(k)}$ for all $j = 1,\dots,n$. Hence, $Av^{(k)} = 2\cos\left(\frac{k\pi}{n+1}\right)v^{(k)}$.

□

Using this proposition we state the following Theorem.

Theorem 5.1.1. *The path graph P_n admits a non-trivial solution to the neighbour-sum problem if and only if $n \equiv 2 \pmod 3$.*

Proof. By Proposition 5.1.1, the eigenvalues of the adjacency matrix of P_n are given by

$$\lambda_k = 2\cos\left(\frac{k\pi}{n+1}\right), \qquad k = 1,\dots,n.$$

By Theorem 4.1.1, the graph P_n admits a non-trivial solution if and only if 1 is an eigenvalue of its adjacency matrix. Therefore, we require $2\cos\left(\frac{k\pi}{n+1}\right) = 1$ for some $k \in \{1,\dots,n\}$. Equivalently, $\cos\left(\frac{k\pi}{n+1}\right) = \frac{1}{2}$.
Since $\cos(\pi/3) = 1/2$, this holds precisely when $\frac{k\pi}{n+1} = \frac{\pi}{3}$. Hence, $n+1 = 3k$.
Therefore, 3 divides $n+1$, which is equivalent to $n \equiv 2 \pmod 3$.

Conversely, suppose that $n \equiv 2 \pmod 3$. Then $n+1 = 3k$ for some integer k. Substituting this value into the eigenvalue formula yields $\lambda_k = 2\cos\left(\frac{k\pi}{n+1}\right) = 2\cos\left(\frac{\pi}{3}\right) = 1$. Thus, 1 is an eigenvalue of the adjacency matrix of P_n. By Theorem 4.1.1, it follows that P_n admits a non-trivial solution to the neighbour-sum problem. □

Remark. Since $1 \le k \le n$, we have $0 < \frac{k\pi}{n+1} < \pi$. The only solution of $\cos(x) = \frac{1}{2}$ in the interval $(0,\pi)$ is $x = \pi/3$. Therefore, $\frac{k\pi}{n+1} = \frac{\pi}{3}$, and hence $n+1 = 3k$.

So by Theorem 5.1.1 solutions to the problem for paths occur for P_2, P_5, P_8, P_{11},…

5.1.1. Example: a solution on P_5

For example, the path graph P_5 admits the solution

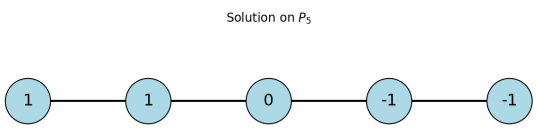

Figure 5.1: path example

Thus P_5 gives a concrete example of a non-trivial solution.

5.2. Cycle graphs

Now we define a cycle graph, and then we investigate when a non-trivial solution exists for cycles.

Definition 5.2.1. *A **cycle graph** C_n is obtained by connecting the endpoints of a path graph, forming a closed loop. Formally, it has vertex set $V(C_n) = \{0, 1, \ldots, n-1\}$, where each vertex i is adjacent to $i+1$ and $i-1$ modulo n.*

Cycle graphs can be viewed as circulant graphs with generating set $S = \{1, -1\}$.

Unlike path graphs, cycle graphs have a fully periodic structure with no boundary vertices. As a result, their adjacency matrices are circulant and their eigenvalues can be derived directly using discrete Fourier methods in Chapter 6.

The adjacency matrix of C_n is $A = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 & 1 \\ 1 & 0 & 1 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \ddots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & 1 & \vdots \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & \cdots & 1 & 0 \end{pmatrix}$.

Proposition 5.2.1. *The eigenvalues of the cycle graph C_n are*

$$\lambda_k = 2\cos\left(\frac{2\pi k}{n}\right), \qquad k = 0, \ldots, n-1.$$

This will be proven in Section 6.3.2.

We now apply this to the neighbour-sum problem.

Theorem 5.2.1. *The cycle graph C_n admits a non-trivial solution to the neighbour-sum problem if and only if $6 \mid n$.*

Proof. By Proposition 5.2.1, the eigenvalues of the adjacency matrix of C_n are

$$\lambda_k = 2\cos\left(\frac{2\pi k}{n}\right), \qquad k = 0, \ldots, n-1.$$

By Theorem 4.1.1, the graph C_n admits a non-trivial solution if and only if 1 is an eigenvalue of its adjacency matrix. Therefore, we require $2\cos\left(\frac{2\pi k}{n}\right) = 1$ for some $k \in \{0, \ldots, n-1\}$. Equivalently, $\cos\left(\frac{2\pi k}{n}\right) = \frac{1}{2}$.

Since $0 \le \frac{2\pi k}{n} < 2\pi$, the solutions of $\cos(\theta) = \frac{1}{2}$ are $\theta = \frac{\pi}{3}$ or $\theta = \frac{5\pi}{3}$. Hence, $\frac{2\pi k}{n} = \frac{\pi}{3}$ or $\frac{2\pi k}{n} = \frac{5\pi}{3}$, which is equivalent to $\frac{k}{n} = \frac{1}{6}$ or $\frac{k}{n} = \frac{5}{6}$. This implies $6 \mid n$.

Conversely, suppose that $6 \mid n$. Then $n = 6k$ for some integer k. Substituting this value into the eigenvalue formula yields $\lambda_k = 2\cos\left(\frac{2\pi k}{n}\right) = 2\cos\left(\frac{2\pi k}{6k}\right) = 2\cos\left(\frac{\pi}{3}\right) = 1$. Thus, 1 is an eigenvalue of the adjacency matrix of C_n.

Therefore, by Theorem 4.1.1, C_n admits a non-trivial solution if and only if n is divisible by 6. □

Remark. Unlike the case of path graphs, the angle $\theta = \frac{2\pi k}{n}$ is not restricted to the interval $(0, \pi)$. Since $k = 0, \ldots, n-1$, we have $0 \le \theta < 2\pi$. Therefore, when solving $\cos(\theta) = \frac{1}{2}$, both solutions on the unit circle must be considered. Hence, $\theta \equiv \pm\frac{\pi}{3} \pmod{2\pi}$.

So by Theorem 5.2.1 non- trivial solutions to the problem for cycles occur for C_6, C_{12}, C_{18}, …

5.2.1. Example: a solution on C_6

As a concrete example, consider the cycle graph C_6. A non-trivial solution to the neighbour-sum problem is given by

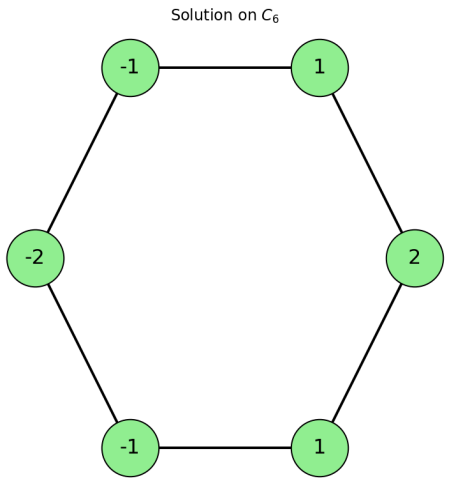

Figure 5.2: cycle example

Indeed, each entry equals the sum of its two neighbours.

Equivalently, if $x = \begin{pmatrix} 2 \\ 1 \\ -1 \\ -2 \\ -1 \\ 1 \end{pmatrix}$, then $Ax = x$, where A is the adjacency matrix of C_6.

This example illustrates concretely how the existence of eigenvalue 1 produces a non-trivial solution to the neighbour-sum problem.

6

Cayley and circulant graphs

In the previous chapter, we obtained complete characterizations of the neighbour-sum problem for path and cycle graphs by explicitly determining the eigenvalues of their adjacency matrices. We now turn to a broader class of graphs, namely Cayley graphs. These graphs arise from finite groups and possess a high degree of symmetry, making them particularly well suited for spectral analysis.

For Cayley graphs over finite abelian groups, the spectrum of the adjacency matrix can be described using Fourier analysis on finite groups. More precisely, the eigenvectors are given by the characters of the underlying group, allowing the eigenvalues to be computed explicitly. This provides a useful strategy for studying the existence of non-trivial solutions to the neighbour-sum problem.

To develop this approach, we first introduce the basic notions of characters and Fourier analysis on finite abelian groups.

6.1. Characters

The material in this section is based on Terras, *Fourier Analysis on Finite Groups and Applications* [5].

Definition 6.1.1. *Let G be a finite abelian group. A **character** of G is a group homomorphism $\chi : G \to \mathbb{C}^\times$, where $\mathbb{C}^\times = \mathbb{C} \setminus \{0\}$ denotes the multiplicative group of nonzero complex numbers.*
Since χ is a group homomorphism, for all $g, h \in G$, $\chi(g+h) = \chi(g)\chi(h)$.

The set of all characters of G is called the *dual group* of G and is denoted by $\widehat{G}$. A fundamental result states that $|\widehat{G}| = |G|$ and that the characters form an orthogonal basis of the vector space $\mathbb{C}^G$ of complex-valued functions on G.

Characters play the role of Fourier modes on finite groups. Just as the functions e^{ikx} form the basic oscillatory modes in classical Fourier analysis, the characters form the basic oscillatory functions on finite abelian groups.

6.2. Cayley graphs

Definition 6.2.1. *Let G be a finite group and let $S \subseteq G \setminus \{e\}$ be a symmetric generating set, i.e., $S = S^{-1}$.*
*The **Cayley graph** $\mathrm{Cay}(G, S)$ is the graph with vertex set G, where two vertices $g, h \in G$ are adjacent if and only if there exists $s \in S$ such that $h = gs$.*

The assumption $S = S^{-1}$ ensures that the graph is undirected.
Cayley graphs are highly symmetric, since the local structure around every vertex is identical. This symmetry allows algebraic techniques to be applied to the study of the adjacency matrix.

Theorem 6.2.1. *Let* $\mathrm{Cay}(G,S)$ *be a Cayley graph over a finite abelian group* G*, and let* A *be its adjacency matrix. Then the eigenvalues of* A *are given by*

$$\lambda_\chi = \sum_{s\in S} \chi(s),$$

where χ *ranges over all characters of* G*. Let* $\widehat{G}$ *denote the set of all characters of* G*. Moreover, each character* $\chi \in \widehat{G}$ *is an eigenvector corresponding to the eigenvalue* λ_χ*.*

Proof. Let $\chi \in \widehat{G}$ be a character. Then $(A\chi)(g) = \sum_{s\in S} \chi(gs)$. Since χ is a homomorphism, $\chi(gs) = \chi(g)\chi(s)$.

Hence, $(A\chi)(g) = \sum_{s\in S} \chi(g)\chi(s)$. Factoring out $\chi(g)$, we obtain $(A\chi)(g) = \chi(g)\sum_{s\in S}\chi(s)$.

Therefore, $(A\chi)(g) = \lambda_\chi \chi(g)$, where $\lambda_\chi = \sum_{s\in S}\chi(s)$.

Since this holds for every $g \in G$, it follows that $A\chi = \lambda_\chi \chi$. Hence, χ is an eigenvector of A with eigenvalue λ_χ. □

Remark. The assumption that G is abelian is essential. For finite abelian groups, the characters form a complete set of one-dimensional representations and hence provide a basis of eigenvectors for the adjacency matrix. For non-abelian groups, one-dimensional characters are generally not sufficient, and higher-dimensional representations are required to describe the spectrum.

Thus, the spectral problem for an abelian Cayley graph reduces to evaluating the characters of the underlying group on the generating set S.

6.3. Circulant graphs

An important special case occurs when $G = \mathbb{Z}_n$. The resulting Cayley graphs are called circulant graphs.

Definition 6.3.1. *Let* $S \subseteq \mathbb{Z}_n$*. The* ***circulant graph*** $\mathrm{Cay}(\mathbb{Z}_n, S)$ *has vertex set* $\{0,1,\dots,n-1\}$*, where vertices* i *and* j *are adjacent if* $j \equiv i + s \pmod{n}$ *for some* $s \in S$*.*

We now formulate the following theorems to give an explicit eigenvalue condition for circulant graphs for the existence of a non-trivial solution to the neighbour-sum problem.

6.3.1. Conditions on eigenvectors and eigenvalues

Theorem 6.3.1. *Every character of* $\mathbb{Z}_n$ *is of the form*

$$\chi_k(j) = e^{2\pi i k j/n},$$

where $k \in \{0,1,\dots,n-1\}$*.*

Proof. Let χ be a character of $\mathbb{Z}_n$. Since $\mathbb{Z}_n$ is generated by 1, the character is completely determined by the value of $\chi(1)$.

Using the homomorphism property, $\chi(a+b) = \chi(a)\chi(b)$, we obtain $\chi(j) = \chi(1)^j$ for every $j \in \mathbb{Z}_n$. Since $\mathbb{Z}_n$ has order n, $n\cdot 1 = 0 \pmod{n}$. Applying χ gives $\chi(n) = \chi(0)$. Since $\chi(n) = \chi(1)^n$ and $\chi(0) = 1$ (identity elements), it follows that $\chi(1)^n = 1$. Therefore, $\chi(1)$ is an nth root of unity, so $\chi(1) = e^{2\pi i k/n}$ for some $k \in \{0,1,\dots,n-1\}$. Substituting this into $\chi(j) = \chi(1)^j$ yields $\chi_k(j) = e^{2\pi i k j/n}$. Therefore, every character of $\mathbb{Z}_n$ has the stated form. □

By Theorem 6.2.1, these characters form the eigenvectors of every circulant graph. Consequently, the eigenvectors are known explicitly.

The following theorem is one of the main results of this chapter. It provides an explicit formula for the spectrum of a circulant graph.

Theorem 6.3.2. *Let* $\mathrm{Cay}(\mathbb{Z}_n, S)$ *be a circulant graph. Then the eigenvalues of its adjacency matrix are*

$$\lambda_k = \sum_{s \in S} e^{2\pi i k s/n}, \qquad k = 0, 1, \ldots, n-1.$$

Proof. By Theorem 6.2.1, the eigenvalues of a Cayley graph are given by $\lambda_\chi = \sum_{s \in S} \chi(s)$, where χ ranges over all characters of the underlying group. Substituting $\chi_k(j) = e^{2\pi i k j/n}$ into this formula gives $\lambda_k = \sum_{s \in S} e^{2\pi i k s/n}$.

This proves the result. □

This is one of the main reasons circulant graphs are particularly suitable for spectral analysis: the eigenvalues can be computed explicitly using Fourier methods.

6.3.2. Solutions for circulant graphs

A non-trivial solution to the neighbour-sum problem corresponds to an eigenvector with eigenvalue 1. Hence, the dimension of the solution space is precisely the multiplicity of the eigenvalue 1.

For circulant graphs over $\mathbb{Z}_n$, this means that we seek integers k such that $\sum_{s \in S} e^{2\pi i k s/n} = 1$. Each such value of k produces a Fourier mode $\chi_k(j) = e^{2\pi i k j/n}$, which gives an eigenvector of the adjacency matrix.

As an illustration, consider the cycle graph $C_6 = \mathrm{Cay}(\mathbb{Z}_6, \{-1, 1\})$.

By Theorem 6.2.1, its eigenvalues are given by $\lambda_k = \sum_{s \in \{-1,1\}} e^{2\pi i k s/6} = e^{2\pi i k/6} + e^{-2\pi i k/6}$.

Using Euler's formula, this simplifies to $\lambda_k = 2\cos\left(\frac{2\pi k}{6}\right)$. Which proves Proposition 5.2.1 in the previous chapter.

By Theorem 4.1.1, the graph admits a non-trivial solution to the neighbour-sum problem if and only if 1 is an eigenvalue. Therefore, we solve $2\cos\left(\frac{2\pi k}{6}\right) = 1$, This occurs for $k = 1$ or $k = 5$. The corresponding characters are $\chi_1(j) = e^{2\pi i j/6}$, and $\chi_5(j) = e^{-2\pi i j/6}$.

These are complex-valued eigenvectors. By Remark 6.3.1, taking the real part of the Fourier mode produces a real-valued eigenvector. Taking real parts yields $\left(1, \frac{1}{2}, -\frac{1}{2}, -1, -\frac{1}{2}, \frac{1}{2}\right)$, which is an eigenvector with eigenvalue 1. Hence, it provides a non-trivial solution to the neighbor-sum problem on C_6. Taking imaginary parts gives $(0, \frac{\sqrt{3}}{2}, \frac{\sqrt{3}}{2}, 0, -\frac{\sqrt{3}}{2}, -\frac{\sqrt{3}}{2})$, which also is a solution to our puzzle. The solutions are shown in Figure 6.1 and Figure 6.2.

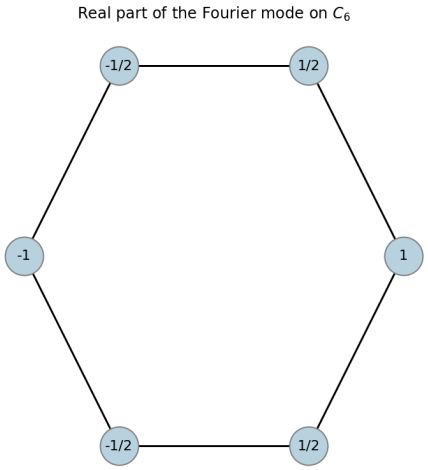

Figure 6.1: Real part solution

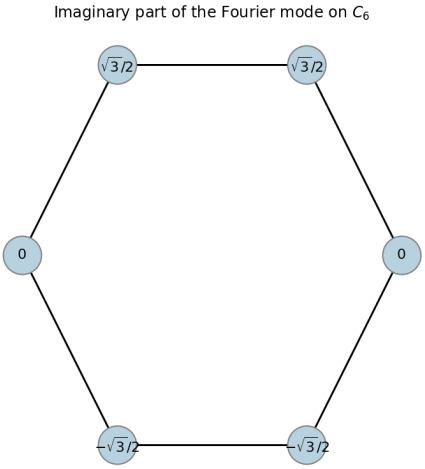

Figure 6.2: Imaginary part solution

Remark 6.3.1. The eigenvectors obtained from the characters of $\mathbb{Z}_n$ are generally complex-valued. Since the adjacency matrix is real and symmetric, the corresponding eigenspaces admit a basis of real eigenvectors. These can be obtained by taking the real and imaginary parts of the complex eigenvectors.

Indeed, suppose $x = u + iv$ is a complex eigenvector of the adjacency matrix A with eigenvalue λ, where u and v denote the real and imaginary parts of x. Since A is a real matrix, we have $A(u+iv) = Au + iAv$.

On the other hand, $Ax = \lambda x = \lambda u + i\lambda v$. Comparing real and imaginary parts gives $Au = \lambda u$ and $Av = \lambda v$. Therefore, both the real and imaginary parts are themselves eigenvectors with eigenvalue λ.

We now consider another example of a circulant graph for which there exists no non-trivial solution.

6.3.3. Example: $\mathrm{Cay}(\mathbb{Z}_8, \{1,3,-1,-3\})$

Consider the circulant graph $\mathrm{Cay}(\mathbb{Z}_8, \{1,3,-1,-3\})$ in Figure 6.3.

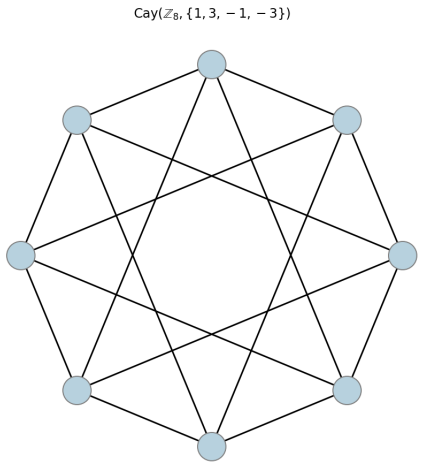

Figure 6.3: $\mathrm{Cay}(\mathbb{Z}_8, \{1,3,-1,-3\})$

The eigenvalues are $\lambda_k = e^{2\pi ik/8} + e^{-2\pi ik/8} + e^{6\pi ik/8} + e^{-6\pi ik/8}$. Using Euler's formula, this simplifies to $\lambda_k = 2\cos\left(\frac{2\pi k}{8}\right) + 2\cos\left(\frac{6\pi k}{8}\right)$.

Evaluating this gives

k	λ_k
0	4
1	0
2	0
3	0
4	-4
5	0
6	0
7	0

.

Thus the spectrum is $\{4,-4,0,0,0,0,0,0\}$. In particular, the eigenvalue 1 does not occur. Hence, the graph $\mathrm{Cay}(\mathbb{Z}_8,\{1,3,-1,-3\})$ does not admit a non-trivial solution to the neighbour-sum puzzle.

This example shows that even for highly symmetric circulant graphs, the existence of a solution is not automatic. The Fourier formula reduces the problem to evaluating the cosine expressions appearing in the eigenvalue formula.

6.3.4. Computational experiments for circulant graphs

In this chapter, we derived an explicit formula for the eigenvalues of circulant graphs. This allows us to determine computationally whether a circulant graph admits a non-trivial solution without explicitly constructing its adjacency matrix.

For every integer $3 \le n \le 31$, all symmetric generating sets $S \subseteq \mathbb{Z}_n \setminus \{0\}$ satisfying $S = -S$ and $\langle S \rangle = \mathbb{Z}_n$ were generated. The first condition ensures that the corresponding Cayley graph is undirected, while the second guarantees that it is connected.

For every generating set S, the eigenvalues were computed using the formula $\lambda_k = \sum_{s \in S} e^{2\pi i k s/n}$, $k = 0,\dots,n-1$. By Theorem 4.1.1, a circulant graph admits a non-trivial solution if and only if one of these eigenvalues equals 1.

Table 6.1 summarizes the percentage of solvable circulant graphs for each value of n.

An interesting observation is that no connected circulant graph of prime order admits a non-trivial solution in our computations. This can be partially explained by the eigenvalue formula $\lambda_k = \sum_{s \in S} \chi_k(s)$.

If $n = p$ is prime, then every non-trivial character is of the form $\chi_k(s) = e^{2\pi i k s/p}$.

Since multiplication by $k \neq 0$ is a permutation of $\mathbb{Z}_p$, each non-trivial eigenvalue is obtained by summing a subset of the p-th roots of unity. Computationally, none of these sums equals 1 for the connected circulant graphs considered, explaining why no non-trivial solutions were found for prime values of n.

Table 6.1: Number and percentage of solvable circulant graphs on n vertices.

n	Total	Solvable	Percentage
3	1	0	0.00%
4	2	0	0.00%
5	3	0	0.00%
6	5	2	40.00%
7	7	0	0.00%
8	12	5	41.67%
9	14	3	21.43%
10	27	5	18.52%
11	31	0	0.00%
12	54	26	48.15%
13	63	0	0.00%
14	119	16	13.45%
15	123	30	24.39%
16	240	78	32.50%
17	255	0	0.00%
18	490	174	35.51%
19	511	0	0.00%
20	990	284	28.69%
21	1015	175	17.24%
22	2015	211	10.47%
23	2047	0	0.00%
24	4020	1755	43.66%
25	4092	150	3.67%
26	8127	793	9.76%
27	8176	1164	14.24%
28	16254	3530	21.72%
29	16383	0	0.00%
30	32607	10782	33.07%
31	32767	0	0.00%

7

Trees

In the previous chapters, we studied the neighbour-sum problem for highly symmetric classes of graphs such as paths, cycles, and circulant graphs. In this chapter, we look at trees. Unlike circulant graphs, trees generally do not possess enough symmetry to allow explicit Fourier-analytic methods. As a result, the problem becomes more combinatorial and structural.

7.1. Basic observations

Definition 7.1.1. *A **tree** is a connected graph containing no cycles.*

Trees form one of the most fundamental classes of graphs. Their spectral behavior differs significantly from that of circulant graphs.

One of the simplest examples of a tree is the path graph P_n. In Chapter 5, we proved that P_n admits a non-trivial solution if and only if $n \equiv 2 \pmod 3$.

This raises the question whether similar structural conditions exist for more general trees.

7.2. Extending paths to trees

The main goal is to compute trees for which the neighbour-sum puzzle is solvable.

One possible approach is to begin with trees for which the neighbour-sum problem is solvable and investigate whether they can be extended while preserving solvability.

For example, consider the path graph P_5, which admits the solution $(1,1,0,-1,-1)$.

Experimentally, one observes that certain vertices can be extended by attaching additional paths while still preserving the existence of a solution. In particular, vertices lying on a subpath of length three often admit extensions.

However, these extensions are more subtle than initially expected. Small local modifications can significantly change the spectrum of the adjacency matrix and may destroy the eigenvalue 1.

7.2.1. Attaching leaves to a path

We now consider a simple way of extending paths to trees. Starting with a path graph P_n for which we have found that the neighbour sum puzzle is solvable, we attach a new leaf to one of its vertices. This operation creates a tree with exactly one branching point if the leaf is attached to an internal vertex.

Recall that a solution of the neighbour-sum puzzle is a vector x satisfying $Ax = x$.

Equivalently, for every vertex v, we require $x_v = \sum_{u \sim v} x_u$.

Suppose G is a graph with a solution x, and let G' be obtained from G by attaching a new leaf w to a vertex u. We ask when the old solution x can be extended to a solution on G' without changing the values on the original graph.

The equation at the new leaf w is $x_w = x_u$.

However, the equation at the vertex u has also changed, because u now has one additional neighbour. In G', the equation at u becomes $x_u = \sum_{v \sim_G u} x_v + x_w$. Since x was already a solution on G, we also still have $_u = \sum_{v \sim_G u} x_v$. Comparing the two equations gives $x_w = 0$. Together with $x_w = x_u$, this implies $x_u = 0$.

Therefore, the original solution can only be extended without changing the old vertex values if the new leaf is attached to a vertex with value zero. In that case, the value of the new leaf must also be zero. This results in the following proposition:

Proposition 7.2.1. *Let G be a graph with a solution x, and let G' be obtained from G by attaching a new leaf w to a vertex u. Then the same solution can be extended to G' if and only if $x_u = 0$. In that case, the extension is given by $x_w = 0$.*

This observation is useful for paths. For example, the path P_5 has the solution

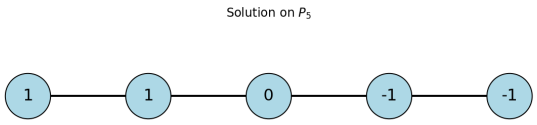

Figure 7.1: path example

The middle vertex has value zero. Therefore, if we attach a new leaf to this middle vertex and give the new leaf value zero, the same solution extends to the new tree:

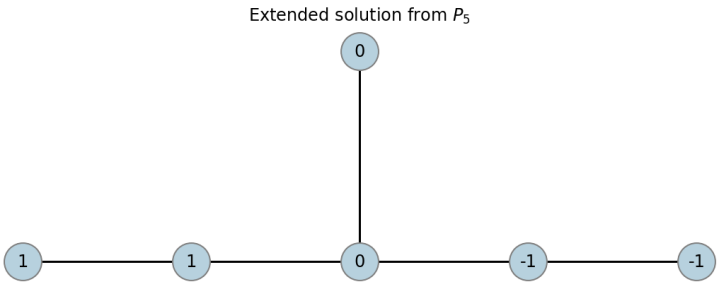

Figure 7.2: Extended path with leaf

On the other hand, attaching a leaf to a vertex with value 1 or -1 cannot preserve this same solution, because the equation at that vertex would gain an extra non-zero contribution.

Thus, attaching leaves to zero-valued vertices gives a way to construct trees with non-trivial solutions from paths. However, this only describes extensions where the old values remain unchanged. It is still possible that after attaching a leaf to a non-zero vertex, the new tree has a different non-trivial solution. This distinction is important: preserving an existing solution is stronger than merely preserving the existence of some solution.

7.2.2. Attaching longer paths

We now extend the previous construction by attaching a path instead of a single leaf.

Let G be a graph with a solution x, and let u be a vertex of G. We construct a new graph G' by attaching a path of length m to u. Denote the new vertices by $y_1, y_2, \ldots, y_m$, where y_1 is adjacent to u, and $y_1 - y_2 - \ldots - y_m$ is a path.

We ask when the old solution on G can be extended to G' without changing any of the values on the original graph.

Proposition 7.2.2. *Let G be a graph with a solution x, and let G' be obtained by attaching a path of length m to a vertex u. If the values on G are kept fixed, then:*

1. *if $x_u = 0$, the path can be attached with all new values equal to zero;*

2. *if $x_u \neq 0$, the path can be attached precisely when $m \equiv 0 \pmod 3$.*

Proof. Since the equation at u was already satisfied in G, the new neighbour y_1 must contribute zero. Hence we must have $y_1 = 0$.

The equations along the attached path are then $y_1 = x_u + y_2$, $y_i = y_{i-1} + y_{i+1}$ for $2 \leq i \leq m-1$, and at the final leaf, $y_m = y_{m-1}$.

If $x_u = 0$, then the simplest extension is obtained by setting $y_1 = y_2 = \cdots = y_m = 0$.

Thus, as in the previous subsection, zero-valued vertices are stable attachment points.

If $x_u \neq 0$, then longer paths can sometimes still be attached without changing the original solution. Since $y_1 = 0$, the equation at y_1 gives $0 = x_u + y_2$, so $y_2 = -x_u$.

The recurrence relation $y_{i+1} = y_i - y_{i-1}$ then determines all later values. This gives the repeating pattern $0, -x_u, -x_u, 0, x_u, x_u, 0, \ldots$ along the attached path.

The final leaf condition $y_m = y_{m-1}$ is satisfied exactly when the attached path ends after a repeated pair. This happens when $m \equiv 0 \pmod 3$.

Therefore, a path of length divisible by 3 can be attached even at a non-zero vertex, provided the values on the new branch are chosen according to the above pattern.

□

This gives a stronger construction method than attaching a single leaf. A single leaf can only be attached to a zero-valued vertex if the old solution is to be preserved. However, a longer branch of length divisible by 3 may also be attached to a non-zero vertex.

For example, consider again the path P_5 with solution If we attach a path of length 3 to the first vertex,

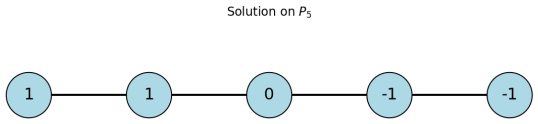

Figure 7.3: path example

which has value 1, then the values on the new branch must be $(0, -1, -1)$.

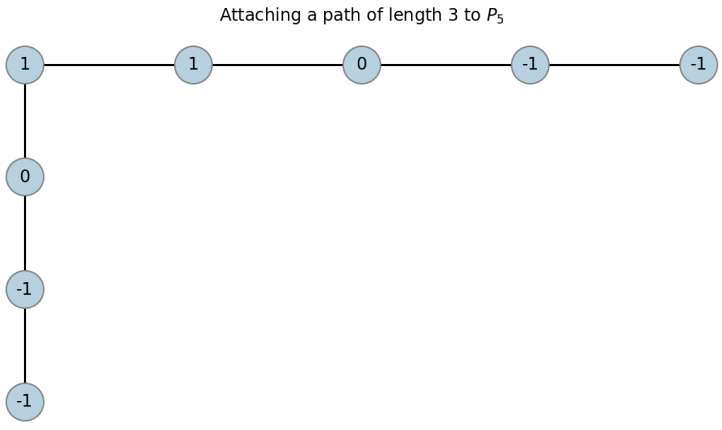

Figure 7.4: Extended path with path

The final two values are equal, so the leaf condition is satisfied. Hence the original solution extends to this larger tree.

This shows that branching does not always destroy a solution. Instead, the length of the attached branch plays an important role. In particular, the same periodic pattern that appears in solutions on paths also appears in branches attached to a tree.

7.2.3. Branching vertices

The constructions in the previous subsections show that paths and attached branches often admit highly structured solutions. In particular, the values along a path follow a periodic pattern. However, the situation becomes more complicated when several branches meet at a single vertex.

Let u be a vertex of degree k. The neighbour-sum equation at u becomes $x_u = x_1 + x_2 + \cdots + x_k$, where $x_1, \ldots, x_k$ are the values on the neighbouring vertices.

For path graphs, every internal vertex has degree 2, so the equation reduces to a simple recurrence relation. This recurrence is responsible for the periodic behaviour observed earlier. At a branching vertex, however, several different branches interact simultaneously, and their contributions must balance exactly.

This creates strong restrictions on possible solutions. For example, suppose a vertex u has value zero. Then the neighbouring values must satisfy $x_1 + x_2 + \cdots + x_k = 0$. Hence positive and negative contributions must cancel.

Similarly, if u has non-zero value, then the sum of all neighbouring values is forced to equal that value. As the degree increases, this balancing condition becomes increasingly restrictive.

Branching vertices impose strong additional constraints on possible solutions. In contrast, paths have minimal branching and therefore admit simple periodic patterns.

More generally, branching vertices impose compatibility conditions between different branches of the tree. Understanding these interactions is one of the main difficulties in analysing trees. We will show some examples in the next section.

7.3. Special families Of trees

In the previous sections, we studied local operations on trees, such as attaching leaves and longer paths. We now consider several specific families of trees. These examples illustrate how branching structure influences the existence of non-trivial solutions.

7.3.1. Star graphs

The star graph $K_{1,n}$ consists of a central vertex connected to n leaves. It is one of the simplest examples of a highly branching tree.

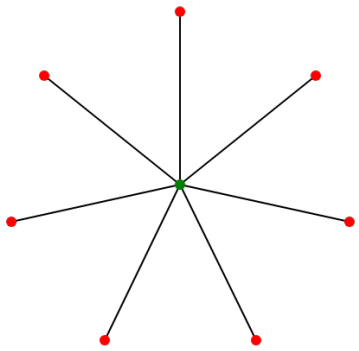

Figure 7.5: Example star graph $K_{1,7}$

Proposition 7.3.1. *The star graph $K_{1,n}$ admits a non-trivial solution if and only if $n = 1$.*

Proof. Let the central vertex have value c, and let the leaves have values $a_1, \ldots, a_n$. Since each leaf is adjacent only to the centre, the neighbour-sum equations at the leaves give $a_i = c$ for all i.

The equation at the central vertex then becomes $c = a_1 + \cdots + a_n = nc$. Hence $(n-1)c = 0$.

If $n > 1$, this implies $c = 0$, and therefore all leaf values are also zero. Thus, star graphs admit only the trivial solution whenever $n > 1$.

□

This example shows that strong branching can impose severe restrictions on possible solutions.

7.3.2. Double stars

A double star is obtained by connecting the centres of two star graphs by an edge. More precisely, let $S(a,b)$ denote the tree obtained from two adjacent vertices u and v, where u has a additional leaves and v has b additional leaves.

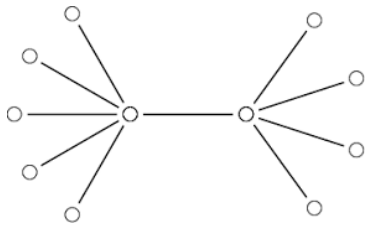

Figure 7.6: Double star $S(5,4)$

Let the values at the central vertices be x_u and x_v. Since every leaf attached to u is adjacent only to u, all these leaves must also have value x_u. Similarly, every leaf attached to v has value x_v.

The neighbour-sum equations at the two central vertices become

$$x_u = ax_u + x_v,$$

$$x_v = bx_v + x_u.$$

Rewriting gives

$$(1-a)x_u = x_v,$$

$$(1-b)x_v = x_u.$$

Substituting one equation into the other yields

$$(1-a)(1-b)x_u = x_u.$$

Hence,

$$\big((1-a)(1-b)-1\big)x_u = 0.$$

If $x_u = 0$, then the equation

$$(1-a)x_u = x_v$$

implies that $x_v = 0$. Since every leaf has the same value as its adjacent central vertex, it follows that all leaf values are also zero. Therefore, the only solution in this case is the trivial solution.

Consequently, a non-trivial solution can exist only if

$$(1-a)(1-b)-1 = 0,$$

or equivalently,

$$(1-a)(1-b) = 1.$$

Proposition 7.3.2. *The double star $S(a,b)$ admits a non-trivial solution if and only if $(a,b) = (2,2)$, apart from the degenerate case $a = b = 0$, which corresponds to the path P_2.*

Proof. From the previous equations, a non-trivial solution can exist only if

$$(1-a)(1-b)=1.$$

Since a and b are non-negative integers, the equation

$$(1-a)(1-b)=1$$

implies that both factors must equal either 1 or -1.

The case

$$1-a=1-b=1$$

gives $a=b=0$, which corresponds to the path P_2.

The non-trivial double star case is therefore

$$1-a=1-b=-1,$$

hence

$$a=b=2.$$

Conversely, if $a=b=2$, then one checks directly that a non-trivial solution exists. □

The graph in Figure 7.7 is an example with $a=b=2$, and thus has a solution for the problem.

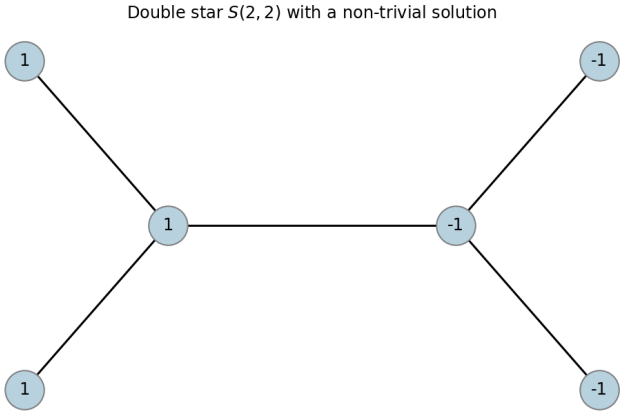

Figure 7.7: Double star S(2,2) solution

7.3.3. Caterpillar trees

Caterpillar trees form an intermediate class between paths and general trees. A caterpillar tree is a tree such that, after removing all leaves, the remaining graph is a path. This remaining path is called the *spine* of the caterpillar.

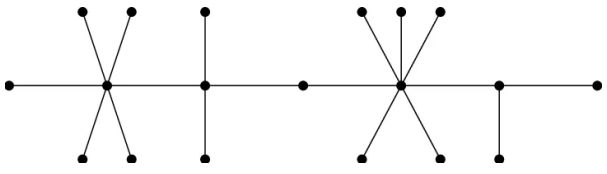

Figure 7.8: Example of a caterpillar tree.

Because the spine is a path, one may expect the periodic behaviour observed for paths to persist. However, every leaf attached to the spine changes the neighbour-sum equation at the corresponding spine vertex.

Suppose the spine consists of the vertices $v_1, v_2, \ldots, v_m$, and suppose that the vertex v_i has n_i attached leaves. Let x_i denote the value assigned to v_i.

Since every attached leaf is adjacent only to v_i, each leaf must also have value x_i. Consequently, the neighbour-sum equation at an interior spine vertex ($2 \le i \le m-1$) becomes

$$x_i = x_{i-1} + x_{i+1} + n_i x_i,$$

or equivalently,

$$(1 - n_i)x_i = x_{i-1} + x_{i+1}.$$

Thus, unlike an ordinary path, the coefficients of the recurrence now depend on the number of leaves attached to each spine vertex.

At the endpoints of the spine, there is only one neighbouring spine vertex. Hence the equations become $x_1 = x_2 + n_1 x_1$, and $x_m = x_{m-1} + n_m x_m$, or equivalently,

$$(1 - n_1)x_1 - x_2 = 0,$$

and

$$-x_{m-1} + (1 - n_m)x_m = 0.$$

Collecting all equations yields the homogeneous linear system

$$M(n_1, \ldots, n_m)x = 0,$$

where

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix},$$

and

$$M(n_1, \ldots, n_m) = \begin{pmatrix} 1-n_1 & -1 & 0 & \cdots & 0 \\ -1 & 1-n_2 & -1 & \cdots & 0 \\ 0 & -1 & 1-n_3 & \ddots & 0 \\ \vdots & \vdots & \ddots & \ddots & -1 \\ 0 & 0 & 0 & -1 & 1-n_m \end{pmatrix}.$$

Proposition 7.3.3. *Let T be a caterpillar tree whose spine vertices have $n_1, \ldots, n_m$ attached leaves. Then T admits a non-trivial solution if and only if* $\det M(n_1, \ldots, n_m) = 0$.

Proof. First suppose that the caterpillar tree admits a non-trivial solution. Restrict this solution to the spine and let $x = (x_1, \ldots, x_m)^T$ denote the values on the spine vertices. Since every leaf is adjacent to exactly one spine vertex, each leaf must have the same value as its adjacent spine vertex. Consequently, the neighbour-sum equations at the spine vertices are precisely the equations encoded by the matrix $M(n_1, \ldots, n_m)x = 0$.

Hence the homogeneous system has a non-zero solution.

Conversely, suppose that the homogeneous system $M(n_1, \ldots, n_m)x = 0$ has a non-zero solution $x = (x_1, \ldots, x_m)^T$.

Assign the value x_i to the spine vertex v_i, and assign the same value x_i to every leaf attached to v_i. Since every leaf has exactly one neighbour, its neighbour-sum equation is automatically satisfied. Moreover, the equations at the spine vertices are exactly those represented by the matrix M. Therefore, this assignment satisfies the neighbour-sum equations on every vertex of the caterpillar, yielding a non-trivial solution.

Finally, a homogeneous linear system has a non-zero solution if and only if its coefficient matrix is singular. Hence, $\det M(n_1, \ldots, n_m) = 0$. □

We have therefore shown that the caterpillar admits a non-trivial solution if and only if the homogeneous system $M(n_1,\dots,n_m)x = 0$ has a non-zero solution.

Proposition 7.3.4. *Let $D_k = \det(M(n_1,\dots,n_k))$ denote the determinant of the leading principal $k\times k$ submatrix of $M(n_1,\dots,n_m)$. Then $D_0 = 1, \qquad D_1 = 1 - n_1$, and for every $k \geq 2$,*

$$D_k = (1 - n_k)D_{k-1} - D_{k-2}.$$

Proof. The initial conditions follow immediately from the definitions. For $k \geq 2$, expand the determinant of $M(n_1,\dots,n_k)$ along the last row. Since the matrix is tridiagonal, only the last two entries contribute, yielding

$$D_k = (1 - n_k)D_{k-1} - D_{k-2}.$$

This is the standard determinant recurrence for tridiagonal matrices [8]. □

The previous proposition shows that the solvability of a caterpillar tree can be determined recursively. Indeed, by Proposition 7.3.3, the caterpillar admits a non-trivial solution if and only if $D_m = 0$. Instead of computing the determinant of the entire matrix directly, it therefore suffices to evaluate the recurrence

$$D_k = (1 - n_k)D_{k-1} - D_{k-2}.$$

This recurrence also allows special families of caterpillar trees to be studied. For example, if every spine vertex has the same number of attached leaves, then the coefficients become constant and the sequence D_k satisfies a second-order linear recurrence with constant coefficients. Such recurrences admit explicit closed-form solutions, making it possible to characterize solvable caterpillar trees within these families. A non-trivial solution of the neighbour-sum problem exists precisely when this homogeneous system has a non-zero solution. By linear algebra, this is equivalent to the coefficient matrix being singular.

Thus, the solvability of a caterpillar tree is completely determined by the numbers of leaves attached to the vertices of its spine. This reduces the neighbour-sum problem on caterpillar trees to studying the determinant of a tridiagonal matrix whose entries depend only on the branching structure.

7.4. Asymptotic solvability of trees

We now relate the local constructions from the previous sections to an asymptotic result of Schwenk [7]. We first define the notion of a limb.

Definition 7.4.1 (Limb)**.** *Let T be a tree and let $v \in V(T)$. A* branch *of T at v is obtained by taking one of the connected components of $T - v$ and adding the vertex v back to it. A **limb** at v is a union of one or more branches at v, viewed as a rooted tree with root v.*

In particular, consider the fixed rooted tree L consisting of a root v and two branches of length two attached to it. Thus L is the rooted tree $a_2 - a_1 - v - b_1 - b_2$, where v is the root. As an unrooted tree, this is simply the path P_5, with the root chosen to be the middle vertex.

Proposition 7.4.1. *If a tree T contains L as a limb, then T admits a non-trivial solution to the neighbour-sum problem.*

Proof. Suppose that T contains the limb L rooted at v. Assign values on this copy of L by

$$x_{a_2} = 1, \qquad x_{a_1} = 1, \qquad x_v = 0, \qquad x_{b_1} = -1, \qquad x_{b_2} = -1.$$

Assign value 0 to all remaining vertices of T.

We check that this gives a solution to the neighbour-sum problem. At the leaves a_2 and b_2, the equations are satisfied because $x_{a_2} = x_{a_1} = 1$ and $x_{b_2} = x_{b_1} = -1$.

At the internal vertices of the two branches, we have $x_{a_1} = x_{a_2} + x_v = 1 + 0 = 1$ and $x_{b_1} = x_{b_2} + x_v = -1 + 0 = -1$.

At the root v, the two non-zero contributions cancel: $x_v = 0 = x_{a_1} + x_{b_1} = 1 + (-1)$.

All vertices outside the limb have value 0, and their neighbours outside the limb also have value 0. If such a vertex is adjacent to the root v, this does not change the equation, since $x_v = 0$. Hence all neighbour-sum equations are satisfied. Since the assignment is not identically zero, T admits a non-trivial solution. □

Schwenk proved that for every fixed limb, the proportion of trees not containing that limb tends to zero as the number of vertices tends to infinity [7]. Applying this result to the fixed limb L above, almost every tree contains L as a limb.

Corollary 7.4.1. *As $n \to \infty$, the proportion of trees on n vertices that admit a non-trivial solution to the neighbour-sum problem tends to* 1.

Proof. Let t_n denote the number of trees on n vertices, and let s_n denote the number of such trees that admit a non-trivial solution. Let b_n be the number of trees on n vertices that do not contain the limb L.

By Schwenk's theorem,

$$\frac{b_n}{t_n} \longrightarrow 0 \qquad \text{as } n \to \infty.$$

Every tree containing L as a limb is solvable by the previous proposition. Therefore, the only possible non-solvable trees are among the trees not containing L. Hence

$$t_n - s_n \leq b_n.$$

Dividing by t_n gives

$$1 - \frac{s_n}{t_n} \leq \frac{b_n}{t_n}.$$

Since the right-hand side tends to 0, it follows that

$$\frac{s_n}{t_n} \longrightarrow 1.$$

Thus, asymptotically almost every tree admits a non-trivial solution to the neighbour-sum problem. □

The asymptotic result shows that the neighbour-sum problem behaves fundamentally differently for large trees than one might expect from small examples. While determining solvability for an individual tree can be difficult, almost every sufficiently large tree admits a non-trivial solution. This provides a theoretical explanation for the increasing proportion of solvable trees observed in the computational experiments of the next chapter.

8

Computational experiments for trees

The theoretical analysis of trees is more difficult than for paths and cycles. Unlike highly symmetric graphs, trees generally do not admit explicit formulas for their eigenvalues or eigenvectors. Therefore, computational methods become a useful tool for exploring possible patterns.

In this chapter, we investigate trees computationally using Python. For each tree, we check if a eigenvalue equal to 1 occurs in the spectrum of the corresponding adjacency matrix. We then compare different structural properties, such as the number of leaves, the maximum degree, and the diameter of the tree.

The dimension of eigenspace for eigenvalue 1 of the trees is also analyzed.

8.1. Computational method

Recall that a graph admits a non-trivial solution to the neighbour-sum problem if and only if its adjacency matrix has eigenvalue 1. Equivalently, if A denotes the adjacency matrix of the graph, then a non-trivial solution exists if and only if $(A-I)x = 0$ has a non-zero solution.

Rather than explicitly computing the eigenvalues of A, we determine whether 1 is an eigenvalue by computing the rank of $A-I$. Indeed, $\operatorname{rank}(A-I) < |V(T)|$ if and only if the homogeneous system has a non-zero solution, and hence if and only if 1 is an eigenvalue of A.

The main goal of the computational experiments in this chapter is to investigate whether certain structural properties of a tree influence the existence of a non-trivial solution. In particular, we study the dependence on the number of vertices, the number of leaves, the maximum degree, and the diameter.

Throughout this chapter, we denote the edge connecting two vertices u and v by uv. Thus, $uv \in E(G)$ means that u and v are adjacent in G.

Definition 8.1.1 (Graph Isomorphism [6])**.** *Two graphs $G = (V(G), E(G))$ and $H = (V(H), E(H))$ are said to be **isomorphic** if there exists a bijection $\phi : V(G) \to V(H)$ such that*

$$uv \in E(G) \quad \Longleftrightarrow \quad \phi(u)\phi(v) \in E(H)$$

for every $u, v \in V(G)$.

Intuitively, isomorphic trees have the same structure and differ only by a relabelling of their vertices. Therefore, it is sufficient to consider one representative from each isomorphism class. For every integer $n \leq 20$, all non-isomorphic trees on n vertices were generated using the `NetworkX` library.

For each generated tree T, we construct its adjacency matrix A and compute the rank of $A-I$. If $\operatorname{rank}(A-I) < |V(T)|$, then $A-I$ has a non-trivial kernel, and hence 1 is an eigenvalue of A. Therefore, the tree is classified as *solvable*. Otherwise, 1 is not an eigenvalue of A, and the tree is classified as *non-solvable*.

The computations were performed in Python using the `NetworkX` and `NumPy` libraries.

8.2. Frequency of solvable trees

We first investigate how frequently trees admit a non-trivial solution to the neighbour-sum problem. For each value of n till $n = 20$, we generated all non-isomorphic trees on n vertices and determined whether the adjacency matrix has eigenvalue 1.

Table 8.1: Frequency solvable trees

n	*total trees*	*solvable trees*	*percentage*
2	1	1	1.000
3	1	0	0.000
4	2	0	0.000
5	3	1	0.333
6	6	2	0.333
7	11	2	0.182
8	23	6	0.261
9	47	14	0.298
10	106	29	0.274
11	235	63	0.268
12	551	166	0.301
13	1301	405	0.311
14	3159	977	0.309
15	7741	2481	0.321
16	19320	6530	0.338
17	48629	16757	0.345
18	123867	43534	0.351
19	317955	115700	0.364
20	823065	308527	0.375

Table 8.1 shows the total number of non-isomorphic trees, the number of solvable trees, and the corresponding proportion.

The results show that solvable trees are considerably more common than initially expected. For example, among the 823,065 non-isomorphic trees on 20 vertices, 308,527 admit a non-trivial solution, corresponding to approximately 37.5% of all trees.

For small values of n, the proportions fluctuate substantially because the total number of trees is very small. Starting from approximately $n = 10$, however, a more stable trend emerges. The proportion of solvable trees increases from about 27% at $n = 10$ to more than 37% at $n = 20$. The data suggests a gradual increase in the proportion of solvable trees.

This observation motivates a more detailed investigation into the structural properties of solvable trees. In the following sections, we examine whether quantities such as the number of leaves, the maximum degree, and the diameter influence the likelihood that a tree admits a non-trivial solution.

8.3. Influence of the number of leaves

The aggregated analysis presented earlier combines trees of different sizes. Since the number of leaves is strongly correlated with the number of vertices, this may obscure the actual influence of the number of leaves on solvability. To separate these effects, we repeated the analysis for several fixed values of n.

For each value $n = 14, \ldots, 18$, all non-isomorphic trees on n vertices are generated. The trees are grouped according to their number of leaves, and for each group we computed the proportion of solvable trees,

$$p_L = \frac{\#\{\text{solvable trees with } L \text{ leaves}\}}{\#\{\text{trees with } L \text{ leaves}\}}.$$

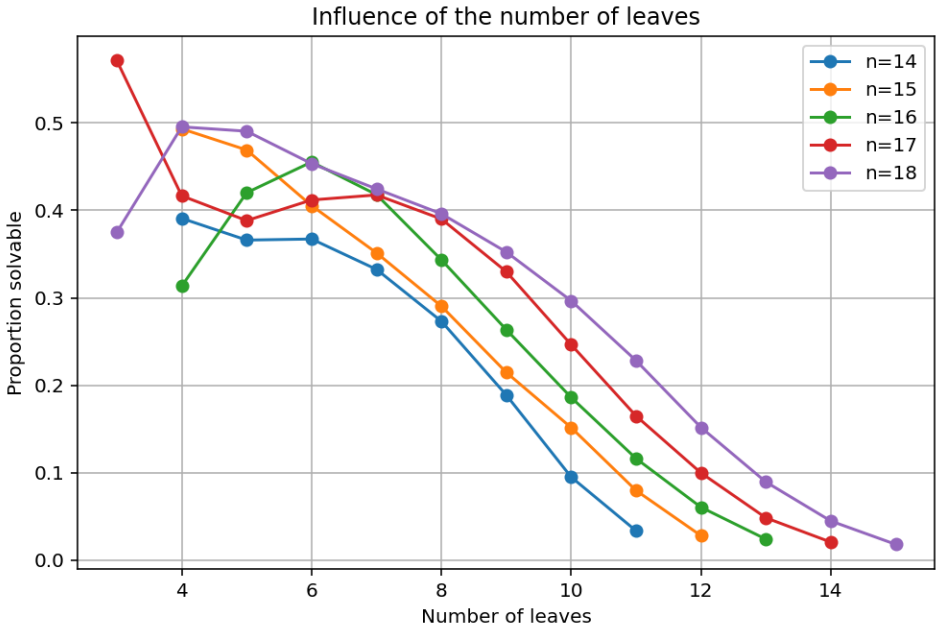

Figure 8.1: Influence of the number of leaves for fixed n

The resulting proportions are shown in Figure 8.1. Figure 8.1 shows that the overall behaviour is remarkably similar for all five values of n. In each case, the proportion of solvable trees is relatively high for trees with a moderate number of leaves and decreases steadily as the number of leaves increases further.

The exact location of the maximum depends on the value of n. For $n = 14, n = 15$ and $n = 18$, the largest proportion occurs for four leaves. For $n = 16$ the maximum shifts to six leaves. For $n = 17$, the highest proportion occurs for trees with three leaves. However, this category contains only 21 non-isomorphic trees, so this peak should be interpreted with caution. For larger sample sizes, all curves attain their highest values for trees with approximately four to seven leaves and then decrease steadily. So all maxima occur for trees with a relatively small to moderate number of leaves.

Once the number of leaves exceeds this range, the proportion of solvable trees decreases consistently for every value of n. For example, for $n = 18$ the solvability decreases from approximately 50% for trees with four or five leaves to below 5% for trees with fourteen leaves. Similar behaviour is observed for the other values of n.

These computational experiments suggest that the number of leaves influences the existence of a non-trivial solution. Trees with a moderate amount of branching appear to be more likely to admit eigenvalue 1, whereas highly branched trees become progressively less likely to be solvable.

The values corresponding to very small numbers of leaves should be interpreted with some care. Only a small number of non-isomorphic trees exist in these categories, causing the associated percentages to fluctuate considerably. Consequently, the most reliable part of the curves is the region where many trees are present, which is roughly after four leaves.

Although the exact position of the maximum depends on n, every curve exhibits the same qualitative behaviour: an initial maximum for a moderate number of leaves followed by a steady decline.

8.4. Influence of the maximum degree

We also investigate the influence of the maximum degree of a tree on the existence of a non-trivial solution. The maximum degree of a tree is defined as the largest degree among all vertices of the tree. Since vertices of high degree connect several branches simultaneously, they impose stronger balancing conditions in the neighbour-sum equations.

For each value $n = 14, \ldots, 18$, all non-isomorphic trees on n vertices were generated. The trees were grouped according to their maximum degree, and for each value Δ we computed the proportion of solvable trees,

$$p_\Delta = \frac{\#\{\text{solvable trees with maximum degree } \Delta\}}{\#\{\text{trees with maximum degree } \Delta\}}.$$

The resulting proportions are shown in Figure 8.2. .

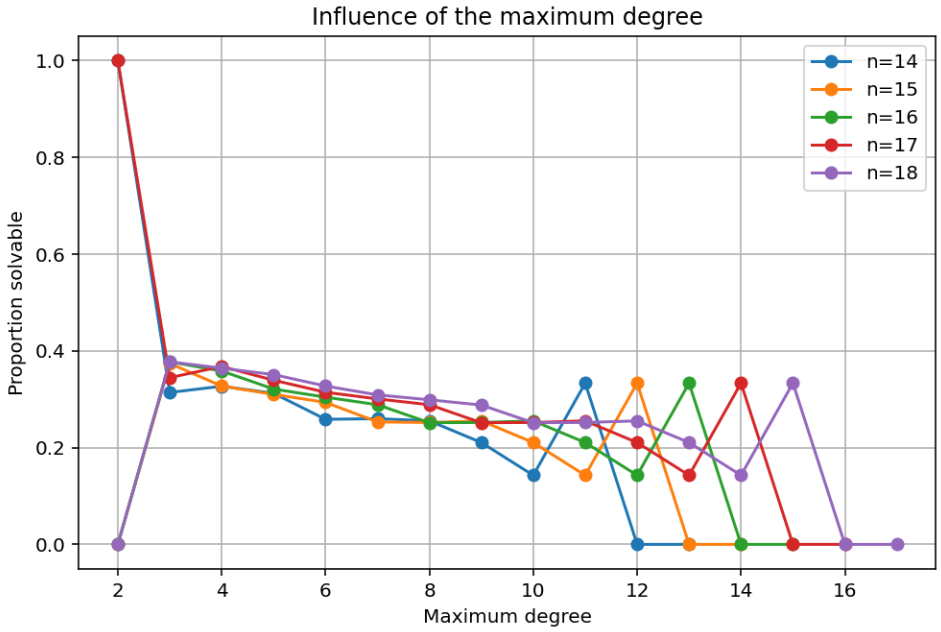

Figure 8.2: Influence of maximum degree

Figure 8.2 shows a similar qualitative behaviour for all considered values of n. For every value of n, the highest proportion of solvable trees occurs for trees with maximum degree three or four. As the maximum degree increases, the proportion of solvable trees gradually decreases.

Although the decrease is less pronounced than for the number of leaves, the same overall trend is visible for all five values of n. For example, when $n = 18$, the proportion decreases from approximately 38% for maximum degree three to approximately 25% for maximum degree ten. The fluctuations for the largest maximum degrees should be interpreted with caution, since only a very small number of non-isomorphic trees exist in these categories. Consequently, the corresponding percentages are strongly affected by the small sample sizes and do not represent a meaningful trend.

Overall, the computational evidence suggests that trees with smaller maximum degree are slightly more likely to admit eigenvalue 1. In contrast, trees containing vertices of very high degree appear to be somewhat less likely to be solvable.

8.5. Influence of the diameter

We next investigate whether the diameter of a tree influences the existence of a non-trivial solution. The diameter of a tree is the length of a longest path between two vertices. Trees with small diameter tend to be highly compact and strongly branched, whereas trees with large diameter are generally more path-like. Since the diameter is strongly correlated with the number of vertices, we again restrict our attention to trees with a fixed number of vertices.

For each value $n = 14, \ldots, 18$, all non-isomorphic trees on n vertices were generated. The trees were grouped according to their diameter, and for each diameter d we computed the proportion of solvable trees,

$$p_d = \frac{\#\{\text{solvable trees with diameter } d\}}{\#\{\text{trees with diameter } d\}}.$$

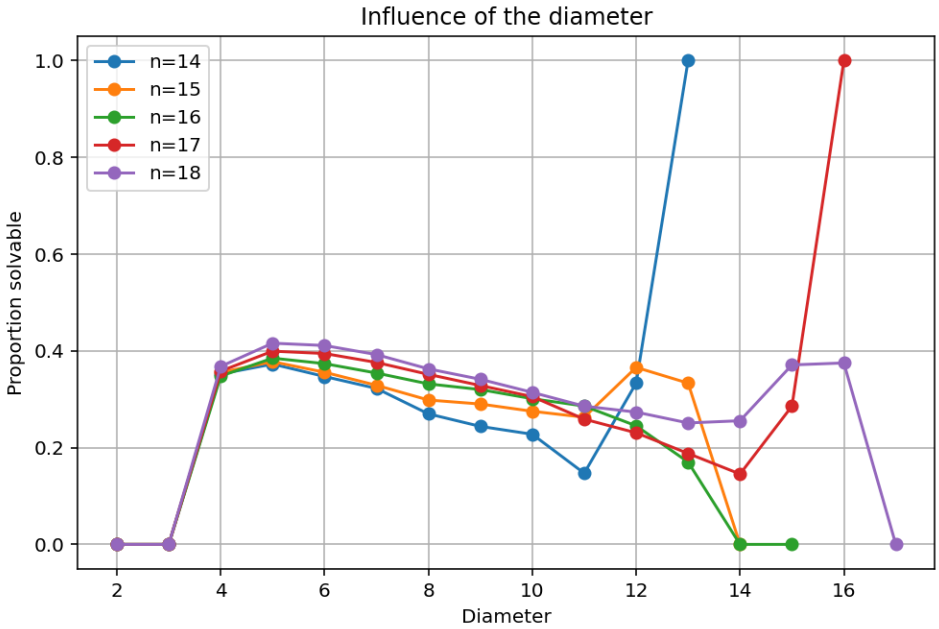

Figure 8.3: Influence of the diameter

The resulting proportions are shown in Figure 8.3.

Figure 8.3 shows a similar qualitative behaviour for all considered values of n. The proportion of solvable trees increases from diameter four to a maximum at diameter five or six, after which it gradually decreases as the diameter increases.

In every case, the proportion of solvable trees increases rapidly from diameter four and reaches its highest value at diameter five. Beyond this point, the proportion of solvable trees decreases gradually as the diameter increases.

Although the exact percentages differ slightly between the various values of n, the overall shape of the curves is nearly identical. For example, the maximum proportion ranges from approximately 37% for $n = 14$ to approximately 42% for $n = 18$, after which a steady decline is observed.

In particular, trees with an intermediate diameter appear to be the most likely to admit a non-trivial solution. By contrast, trees with larger diameters become progressively less likely to be solvable. For example, for $n = 18$ the proportion of solvable trees decreases from approximately 42% for diameter five to approximately 27% for diameter twelve.

The fluctuations observed for the largest diameters should be interpreted with caution, since only a small number of non-isomorphic trees exist in these categories. Consequently, the corresponding percentages are dominated by small sample sizes rather than reflecting a genuine structural phenomenon.

Overall, these computational experiments suggest that the diameter influences the existence of a non-trivial solution. Trees with an intermediate diameter appear to be more likely to admit eigenvalue 1 than trees with either very small or very large diameters.

8.6. Dimension of the solution space

So far, we have only investigated whether a tree admits a non-trivial solution. We now consider the dimension of the corresponding solution space.

Recall that a tree is solvable if and only if the adjacency matrix A has eigenvalue 1. Equivalently, $(A-I)x = 0$ admits a non-trivial solution. The dimension of the solution space is therefore given by $\dim\ker(A-I)$, which equals the multiplicity of the eigenvalue 1.

For every non-isomorphic tree with at most twenty vertices, we computed the dimension of $\ker(A-I)$. The results are summarized in Table 8.2.

Table 8.2: Distribution of solution space dimensions.

Dimension	Number of trees
0	850828
1	360122
2	104858
3	24864
4	4412
5	830
6	90
7	17
8	2

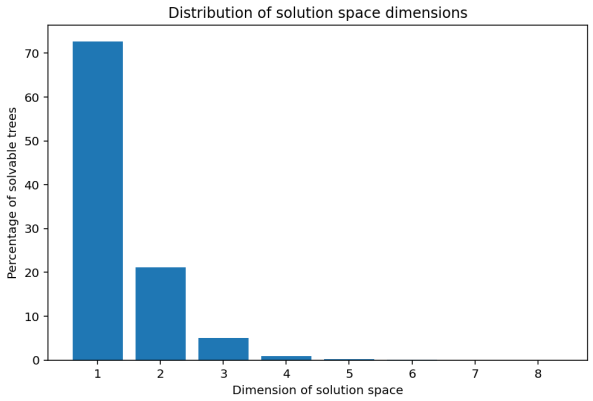

Figure 8.4: Distribution dimensions of solvable trees

As can been seen in Figure 8.4, the vast majority of solvable trees have a one-dimensional solution space. Among all solvable trees, approximately 73% have dimension 1, while about 21% have dimension 2. Dimensions greater than two occur much less frequently, and dimensions larger than five are extremely rare.

The results show that although most solvable trees possess only a single independent family of solutions, higher-dimensional solution spaces are by no means exceptional. In particular, more than one fifth of all solvable trees admit at least two linearly independent solutions.

The frequency decreases rapidly as the dimension increases. Only ninety trees with at most twenty vertices have a six-dimensional solution space, seventeen have dimension seven, and only two examples attain dimension eight. This indicates that large multiplicities of the eigenvalue 1 are rare among trees.

To gain some insight into the occurrence of large solution spaces, we examined the two trees with the largest observed dimension, 8. Both examples are shown in Figure 8.5.

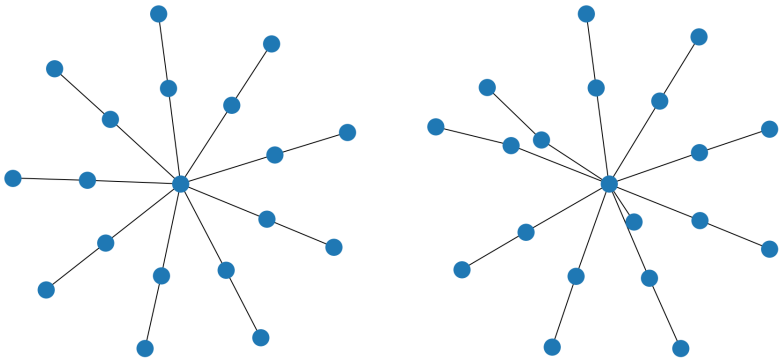

Figure 8.5: The two trees with solution space dimension 8.

Interestingly, the two trees have a very similar structure. Each consists of a central vertex connected to many branches of length two, together with a single shorter branch.

This structure is closely related to the analysis of paths. Any two branches of length two, together with the central vertex, form a copy of the path P_5, for which we showed that a non-trivial solution exists. In the periodic solution $(1,1,0,-1,-1)$, the middle vertex of the path has value 0. Since the central vertex is shared by many such copies of P_5, several local solution patterns can coexist while agreeing on the value at the central vertex. This provides a explanation for the unusually large dimension of the solution space.

9

Conclusion and discussion

In this thesis we studied the neighbour-sum problem on graphs, where the goal is to determine whether there exists a non-trivial assignment of values to the vertices such that every vertex equals the sum of the values assigned to its neighbours. By reformulating the problem as the eigenvalue equation $Ax = x$, the existence of a solution was shown to be equivalent to the adjacency matrix having eigenvalue 1.

For highly symmetric graph classes, explicit characterizations were obtained. For path graphs, a non-trivial solution exists precisely when $n \equiv 2 \pmod 3$, while cycle graphs admit a solution exactly when $6 \mid n$. These results were generalized to Cayley graphs over finite abelian groups using Fourier analysis, leading to an explicit expression for the eigenvalues of circulant graphs in terms of the generating set.

For trees, explicit spectral formulas are generally unavailable. Instead, several theoretical results were obtained for special families of trees. In particular, complete characterizations were derived for star graphs and double stars, while for caterpillar trees the problem was reduced to determining when a tridiagonal matrix is singular. Furthermore, using a theorem of Schwenk, it was shown that asymptotically almost every tree admits a non-trivial solution.

The computational experiments provided additional insight into the structure of solvable trees. The proportion of solvable trees appears to increase with the number of vertices, supporting the asymptotic result. Moreover, trees with a moderate number of leaves, moderate maximum degree, and intermediate diameter were consistently more likely to admit eigenvalue 1 than highly branched or extremely path-like trees. Finally, although most solvable trees have a one-dimensional solution space, examples with much larger dimensions were found, suggesting that repeated local structures may significantly increase the multiplicity of the eigenvalue 1.

Several interesting questions remain open. The computational experiments indicate clear structural patterns, but a theoretical explanation is still lacking. In particular, it would be interesting to characterize solvable trees purely in terms of their branching structure, or to explain why trees with intermediate diameter and a moderate number of leaves are most likely to admit eigenvalue 1. Another promising direction is the determinant recurrence for caterpillar trees. Since this recurrence has constant coefficients for several natural subclasses, explicit closed-form solutions may lead to complete classifications of solvable caterpillar trees. Finally, understanding which local structures give rise to high-dimensional solution spaces may provide further insight into the spectral properties of trees.

Appendix

Python code circulant graphs

```python
import numpy as np
from itertools import combinations
from math import gcd
from functools import reduce

def symmetric_generating_sets(n):

    pairs = []
    used = set()

    for s in range(1, n):
        if s in used:
            continue
        pair = {s, (-s) % n}
        used.update(pair)
        pairs.append(pair)

    all_sets = []
    for r in range(1, len(pairs) + 1):
        for chosen in combinations(pairs, r):
            S = set().union(*chosen)

            # S generates Z_n iff gcd(n, all elements of S) = 1
            g = reduce(gcd, [n] + list(S))
            if g == 1:
                all_sets.append(sorted(S))

    return all_sets


def circulant_eigenvalues(n, S):
    """
    Eigenvalues of Cay(Z_n,S):
    lambda_k = sum_{s in S} exp(2*pi*i*k*s/n).
    """
    eigenvalues = []
    for k in range(n):
        lam = sum(np.exp(2j * np.pi * k * s / n) for s in S)
        eigenvalues.append(lam)
    return np.array(eigenvalues)


def is_solvable_circulant(n, S, tol=1e-8):
    eigenvalues = circulant_eigenvalues(n, S)
    return np.any(np.abs(eigenvalues - 1) < tol)


def experiment(N=31):
    data = []

    for n in range(3, N + 1):
        sets = symmetric_generating_sets(n)
```

```
        total = len(sets)

        solvable = 0
        for S in sets:
            if is_solvable_circulant(n, S):
                solvable += 1

        percentage = 100 * solvable / total if total > 0 else 0

        data.append({
            "n": n,
            "total": total,
            "solvable": solvable,
            "non_solvable": total - solvable,
            "percentage": percentage
        })

    return data


data = experiment(31)

for row in data:
    print(
        f"n={row['n']:2d}: "
        f"total={row['total']:5d}, "
        f"solvable={row['solvable']:5d}, "
        f"percentage={row['percentage']:6.2f}%"
    )
```

Python code frequency trees

```
    import networkx as nx
import numpy as np
import time


def has_solution(T, tol=1e-9):
    A = nx.to_numpy_array(T)

    eigenvalues = np.linalg.eigvals(A)

    return any(abs(lam - 1) < tol for lam in eigenvalues)


print("n | total trees | solvable trees | percentage")
print("-" * 50)

for n in range(2, 21):

    total = 0
    solvable = 0

    for T in nx.nonisomorphic_trees(n):

        total += 1

        if has_solution(T):
            solvable += 1

```

```
    percentage = solvable / total

    print(
        f"{n:2d} | "
        f"{total:8d} | "
        f"{solvable:8d} | "
        f"{percentage:.3f}"
    )
```

Python code for leaves of tree

```
    import networkx as nx
import numpy as np
import matplotlib.pyplot as plt

def solvable(T):
    A = nx.to_numpy_array(T)
    M = A - np.eye(len(T))
    return np.linalg.matrix_rank(M) < len(T)

results = {}

for n in [14, 15, 16, 17, 18]:

    data = {}

    for T in nx.nonisomorphic_trees(n):

        leaves = sum(
            1 for v in T.nodes()
            if T.degree(v) == 1
        )

        if leaves not in data:
            data[leaves] = [0, 0]

        data[leaves][0] += 1

        if solvable(T):
            data[leaves][1] += 1

    x = []
    y = []

    for leaves in sorted(data):

        total, solvable_count = data[leaves]

        x.append(leaves)
        y.append(solvable_count / total)

    results[n] = (x, y)

    print(f"\nn = {n}")
    print("leaves | total | solvable | percentage")
    print("-------------------------------------")

    for leaves in sorted(data):
        total, solvable_count = data[leaves]

```

```
        print(
            leaves,
            "|",
            total,
            "|",
            solvable_count,
            "|",
            round(solvable_count / total, 3)
        )

plt.figure(figsize=(8,5))

for n in [14, 15, 16, 17, 18]:
    x, y = results[n]
    plt.plot(x, y, marker='o', label=f"n={n}")

plt.xlabel("Number of leaves")
plt.ylabel("Proportion solvable")
plt.title("Influence of the number of leaves")
plt.legend()
plt.grid(True)

plt.show()
```

Python code for max degree of tree

```
    import networkx as nx
import numpy as np
import matplotlib.pyplot as plt

def solvable(T):
    A = nx.to_numpy_array(T)
    M = A - np.eye(len(T))
    return np.linalg.matrix_rank(M) < len(T)

results = {}

for n in [14, 15, 16, 17, 18]:

    data = {}

    for T in nx.nonisomorphic_trees(n):

        max_degree = max(dict(T.degree()).values())

        if max_degree not in data:
            data[max_degree] = [0, 0]

        data[max_degree][0] += 1

        if solvable(T):
            data[max_degree][1] += 1

    x = []
    y = []

    for d in sorted(data):
        total, solvable_count = data[d]
        x.append(d)
        y.append(solvable_count / total)
```

```
    results[n] = (x, y)

    print(f"\nn = {n}")
    print("max degree | total | solvable | percentage")
    print("-------------------------------------------")

    for d in sorted(data):
        total, solvable_count = data[d]

        print(
            d,
            "|",
            total,
            "|",
            solvable_count,
            "|",
            round(solvable_count / total, 3)
        )

plt.figure(figsize=(8,5))

for n in [14, 15, 16, 17, 18]:
    x, y = results[n]
    plt.plot(x, y, marker='o', label=f"n={n}")

plt.xlabel("Maximum degree")
plt.ylabel("Proportion solvable")
plt.title("Influence of the maximum degree")
plt.legend()
plt.grid(True)

plt.show()
```

Python code for diameter of tree

```
    import networkx as nx
import numpy as np
import matplotlib.pyplot as plt

def solvable(T):
    A = nx.to_numpy_array(T)
    M = A - np.eye(len(T))
    return np.linalg.matrix_rank(M) < len(T)

results = {}

for n in [14, 15, 16, 17, 18]:

    data = {}

    for T in nx.nonisomorphic_trees(n):

        diameter = nx.diameter(T)

        if diameter not in data:
            data[diameter] = [0, 0]

        data[diameter][0] += 1

```

```
        if solvable(T):
            data[diameter][1] += 1

    x = []
    y = []

    print(f"\nn = {n}")
    print("diameter | total | solvable | percentage")
    print("-----------------------------------------")

    for diameter in sorted(data):
        total, solvable_count = data[diameter]
        percentage = solvable_count / total

        x.append(diameter)
        y.append(percentage)

        print(
            diameter,
            "|",
            total,
            "|",
            solvable_count,
            "|",
            round(percentage, 3)
        )

    results[n] = (x, y)

plt.figure(figsize=(8,5))

for n in [14, 15, 16, 17, 18]:
    x, y = results[n]
    plt.plot(x, y, marker='o', label=f"n={n}")

plt.xlabel("Diameter")
plt.ylabel("Proportion solvable")
plt.title("Influence of the diameter")
plt.legend()
plt.grid(True)

plt.show()
```

Bibliography

[1] *Dutta, S., Mandal, A., Gupta, S., & Chatterjee, S. (2023, 6 oktober). Neighbour Sum Patterns: Chessboards to Toroidal Worlds. arXiv.org.* https://arxiv.org/abs/2310.04401

[2] *Martin-Hernandez, J., Trajanovski, S., Wang, H., Li, C., & Van Mieghem, P. (2012). Zero and non-zero eigenvector components graph matrices. Delft University of Technology.* https://www.nas.ewi.tudelft.nl/people/Stojan/pub/Sign_eigenvectors_adjacency_matrix.pdf

[3] *Devadatta Kulkarni, Darrell Schmidt, and Sze-Kai Tsui. Eigenvalues of tridiagonal pseudo-toeplitz matrices. Linear Algebra and its Applications, 297:63–80, 1999.* https://www.sciencedirect.com/science/article/pii/S0024379599001147

[4] *Chris Godsil, Gordon Royle - Algebraic Graph Theory* https://aeb.win.tue.nl/2WF02/spectra.pdf

[5] *Terras A. Fourier Analysis on Finite Groups and Applications. Cambridge University Press; 1999.* https://doi.org/10.1017/CBO9780511626265f

[6] *Reinhard Diestel. Graph Theory. 5th ed. Springer, 2017.* https://doi.org/10.1007/978-3-662-53622-3

[7] *Allen J. Schwenk. Almost all trees are cospectral. In: New Directions in the Theory of Graphs, Academic Press, 1973.*

[8] *Horn, R. A., and Johnson, C. R.* Matrix Analysis. 2nd edition. Cambridge University Press; 2012. https://www.scribd.com/document/620633226/Roger-a-Horn-Charles-R-Johnson-Topics-in-Matrix-An

[9] *West, D. B.* Introduction to Graph Theory. 2nd edition. Prentice Hall; 2001.