6-2013

Solving Hard Graph Problems with Combinatorial Computing and Optimization

Alexander R. Lange

Solving Hard Graph Problems with Combinatorial Computing and Optimization

by

Alexander R. Lange

A Thesis Submitted
in
Partial Fulfillment of the
Requirements for the Degree of
Master of Science
in
Computer Science

Supervised by

Dr. Stanisław Radziszowski

Department of Computer Science

B. Thomas Golisano College of Computing and Information Sciences
Rochester Institute of Technology
Rochester, New York

June 2013

The thesis "Solving Hard Graph Problems with Combinatorial Computing and Optimization" by Alexander R. Lange has been examined and approved by the following Examination Committee:

Dr. Stanisław Radziszowski
Professor
Thesis Committee Chair

Dr. Ivona Bezáková
Associate Professor
Reader

Dr. Darren A. Narayan
Professor
Observer

Dedication

This thesis is dedicated to Blair Phillips, a truly remarkable and special human being whom I was fortunate enough to call my good friend for the better part of my life. I have looked up to you since I was eight years old, and I know I speak for an uncountable number of people when I say there are an uncountable number of ways I am thankful to have known you.

I recall the look on your face from the night I eagerly explained to you this cool math I just learned called Ramsey theory. It was a look of curiosity, of admiration, of excitement; a look that was as native to you as your red hair. I am thankful that you were there when all of this started, and I look forward to you being a part of it until the end.

Acknowledgments

I am fortunate to owe many people many thanks for who and where I am today.

First, I thank my family and friends for their constant stream of support. I give special thanks to my parents for always being there, for pushing me, and for instilling in me my enjoyment of learning. I thank my brother, Jake, for reminding me to still have fun. I also thank: the Avon group for the enduring foundation; Eli and Matt for assuring my academic interests; Pete for the emergency IT support; Tori for saying it was "really cool" the first night I was too busy with work; Harwin and BTZ for the nights of needed distraction; and Izzy for hanging out on the many nights of no distraction.

The Mathematics and Computer Science faculty and staff have been great to me. I thank Professor Narayan for being on my thesis committee, and for welcoming me into the REU these past two summers. I thank Ivona for reading my thesis, and especially thank her and Edith for the numerous opportunities I have been given to appreciate, explore, and build my confidence in many areas of computer science theory.

I did not accomplish this work on my own. I thank Ivan Livinsky and Xiaodong Xu for collaborating with me, as well as Yongqi Sun and my many peers at RIT. I owe many thanks to Gurcharan Khanna and Research Computing at RIT for the valuable and helpful support, as well as Mats Rynge for his guidance throughout my use of the Open Science Grid.

Last and far from least I thank Staszek for his mentorship, including his distinctive and persistent confidence, patience, and integrity. I am forever indebted to him not only for the many opportunities he has presented me, but for introducing me to a world of mathematics and computer science which I will always hold close.

Contents

Abstract

Many problems arising in graph theory are difficult by nature, and finding solutions to large or complex instances of them often require the use of computers. As some such problems are **NP**-hard or lie even higher in the polynomial hierarchy, it is unlikely that efficient, exact algorithms will solve them. Therefore, alternative computational methods are used. Combinatorial computing is a branch of mathematics and computer science concerned with these methods, where algorithms are developed to generate and search through combinatorial structures in order to determine certain properties of them. In this thesis, we explore a number of such techniques, in the hopes of solving specific problem instances of interest.

Three separate problems are considered, each of which is attacked with different methods of combinatorial computing and optimization. The first, originally proposed by Erdős and Hajnal in 1967, asks to find the Folkman number $F_e(3,3;4)$, defined as the smallest order of a K_4-free graph that is not the union of two triangle-free graphs. A notoriously difficult problem associated with Ramsey theory, the best known bounds on it prior to this work were $19 \leq F_e(3,3;4) \leq 941$. We improve the upper bound to $F_e(3,3;4) \leq 786$ using a combination of known methods and the Goemans-Williamson semi-definite programming relaxation of MAX-CUT. The second problem of interest is the Ramsey number $R(C_4, K_m)$, which is the smallest n such that any n-vertex graph contains a cycle of length four or an independent set of order m. With the help of combinatorial algorithms, we determine $R(C_4, K_9) = 30$ and $R(C_4, K_{10}) = 36$ using large-scale computations on the Open Science Grid. Finally, we explore applications of the well-known Lenstra-Lenstra-Lovász (LLL) algorithm, a polynomial-time algorithm that, when given a basis of a lattice, returns a basis for the same lattice with relatively short vectors. The main result of this work is an application to graph domination, where certain hard instances are solved using this algorithm as a heuristic.

Chapter 1

Introduction

1.1 Overview

This thesis is concerned with problems arising in graph theory, and how the use of computations can assist in determining their solutions. The problems of interest are those which are difficult by nature, and are often associated with problems known to be **NP**-complete. It is therefore unlikely to be able to develop efficient, exact algorithms to solve them, and alternative computational methods are used instead. Combinatorial computing is a branch of mathematics and computer science concerned with such methods, where algorithms are invented and implemented to generate, enumerate and search through combinatorial structures. In this thesis, we study and implement techniques for attacking instances of such problems which are too large to solve "by hand." Some success is obtained, as summarized below.

A number of problems studied in this thesis fall under the branch of mathematics known as Ramsey theory. This subject is primarily concerned with the properties certain mathematical structures need in order to guarantee that desired sub-structures are contained within them. It is often seen as the study of the order that can be derived from chaos. Graphs are combinatorial objects that Ramsey theory is regularly associated with.

A classical problem used to introduce Ramsey theory involves a party in which some people are mutual acquaintances and all others are mutual strangers. The problem asks to determine the minimum number of people needed at such a party so that either three people all know each other or three people all don't know each other. It is straightforward to represent this as a graph problem: Let each person be represented as a vertex and let all vertices be connected to each other, that is, let the graph be *complete*. We will color each edge *red* (*blue*) if the two corresponding people know (don't know) each other. The question

is then to find the minimum n such that the complete graph K_n, when colored this way, always contains either a red or blue triangle. This n is the Ramsey number $R(3,3)$ and it is known that $R(3,3) = 6$.

In 1967, Erdős and Hajnal [40] posed the question: Does there exist a K_4-free graph that is not the disjoint union of two triangle-free graphs? This question is similar to the previous, but instead of coloring the edges of the complete graph, we color those of a K_4-free graph. In 1970, Folkman [49] proved that they indeed exist, and they are now called Folkman graphs. The problem then became to determine how small such a graph could be, known as the Folkman number $F_e(3,3;4)$. The difficulty of this task is apparent, as the best known bounds prior to this work were $19 \leq F_e(3,3;4) \leq 941$. In the first part of this thesis, we employ computational techniques to improve the upper bound to 786. These techniques combine known methods with a novel use of the Goemans-Williamson MAX-CUT semidefinite programming relaxation, as described in Chapter 2.

Analogous Ramsey-type problems exist for graphs different than triangles. The Ramsey number $R(G,H)$ is the smallest n such that for every two-coloring of the edges of K_n, a monochromatic copy of G or H exists in the first or second color, respectively. A main combinatorial computing approach in determining $R(G,H)$ is to computationally construct colorings of K_t that do not contain copies of G in the first color or copies of H in the second, thus establishing $R(G,H) > t$. If a complete enumeration of such colorings is possible, we can determine $R(G,H)$ exactly. Chapter 3 is concerned with such an approach for $R(C_4, K_m)$, where C_4 is the cycle on four vertices. With the use of massive computations on the Open Science Grid, we determine $R(C_4, K_9) = 30$ and $R(C_4, K_{10}) = 36$.

A main goal of this thesis was to apply known techniques in lattice basis reduction to new combinatorial problems, especially those associated with Ramsey theory. The Lenstra-Lenstra-Lovász (LLL) algorithm is a well-known, polynomial-time algorithm that, when given a basis of a lattice, returns a *reduced* basis for the same lattice containing the shortest vectors it can find. Its main application to combinatorial computing involves representing a search problem in a particular matrix form, so that if the column vectors of the matrix are treated as a basis of a lattice, the solution to the problem is found as a short vector of a reduced basis of that lattice.

Although no substantial application to Ramsey or Folkman numbers was found, we had

some success in applying basis reduction to another area of graph theory: *graph domination*. A classical problem associated with this area is one involving the game of chess. The question, originally studied by de Jaenisch in 1862 [35], asks: What is the minimum number of queens needed, so that if placed on an $n \times n$ chessboard, a piece on any other position is capturable by a queen? This problem, like the party problem, can be formulated with a graph. Let each position of a chessboard be a vertex and let two vertices be connected if a queen on the first position can reach the second position in one move. The question is then to find the smallest collection of vertices such that all other vertices are connected to at least one of the collection. This is the problem of graph domination. Chapter 4 presents a method that uses the LLL Algorithm to find *dominating sets* of a graph and presents experimental data exhibiting some success.

As previously mentioned, the common thread of this work is the use of computations in solving specific instances of graph problems which are believed to be too large to be solved by hand. Before discussing the details of these computations in Section 1.3, we now introduce the concepts and notation used throughout this work.

1.2 Background and Notation

Graph Theory

The main theme common to all parts of this thesis is that of graph theory. A *graph* G is a set of *vertices* and *edges* where edges are unordered pairs of vertices. $V(G)$ and $E(G)$ are the vertex set and edge set of G, respectively. If $\{u, v\} \in E(G)$ then u and v are *adjacent* or *connected* vertices, and both u and v are *incident* to edge $\{u, v\}$. All graphs in this work are *loopless* (for any $v \in V(G)$, $\{v, v\} \notin E(G)$) and *unweighted*. The *order* of G is $|V(G)|$ and the *size* of G is $|E(G)|$. The *complement* of G, denoted $\overline{G}$, is defined as $V(\overline{G}) = V(G)$ and $E(\overline{G}) = \{\{u, v\} \mid \{u, v\} \notin E(G) \text{ and } u \neq v\}$. A subgraph of G is a graph H such that $V(H) \subseteq V(G)$ and $E(H) \subseteq \{\{u, v\} |\ u, v \in V(H),\ \{u, v\} \in E(G)\}$. A *directed* graph or *digraph* is a graph in which the edge set is ordered, that is, (u, v) and (v, u) are distinct edges. A *bipartite* graph is a graph whose vertices can be split into two parts such that no two vertices in the same part are adjacent. A *circulant* graph C on n vertices is defined as $V(C) = \mathbb{Z}_n$ and $E(C) = \{\{u, v\} || u - v| \in D\}$ where D is some predefined subset of $\mathbb{Z}_n$.

The *neighborhood* of $v \in V(G)$ is the set of vertices adjacent to v, and is denoted $N_G(v) = \{u \mid \{u, v\} \in E(G)\}$. The *closed neighborhood* of v is $N_G[v] = N_G(v) \cup \{v\}$. The *degree* of v is $\deg_G(v) = |N_G(v)|$. The minimum and maximum degrees of vertices in G are denoted $\delta(G)$ and $\Delta(G)$, respectively. G is *d-regular* if $\deg_G(v) = d$ for all $v \in V(G)$. A subgraph H of G is an *induced* subgraph if $E(H) = \{\{u, v\} |\ u, v \in V(H),\ \{u, v\} \in E(G)\}$, that is, H has all of the edges G has over $V(H)$. If $S = V(H)$, we say H is *induced* by S and write $H = G[S]$. The *join* of two graphs $G_1 + G_2 = G$ is the union of G_1 and G_2 with every vertex of G_1 connected to every vertex of G_2, that is, $V(G) = V(G_1) \cup V(G_2)$ and $E(G) = E(G_1) \cup E(G_2) \cup \{\{g_1, g_2\} \mid g_1 \in V(G_1),\ g_2 \in V(G_2)\}$.

We use common notation to represent important types of graphs that appear throughout this work: K_n is the *complete* graph on n vertices, where every pair of vertices is connected; $K_{s,t}$ is the *complete bipartite* graph, a bipartite graph with parts of order s and t, where each vertex of one part is connected to all of the other; C_n is the n-vertex *cycle* graph, consisting of only a simple cycle; P_n is an n-vertex *path* graph, consisting of only a simple path; W_n is the *wheel* graph, defined as $K_1 + C_{n-1}$; and S_t is the *star* graph, defined as $K_{1,t}$.

We now introduce a number of classical graph properties whose related problems and parameters are explored throughout this thesis. A *clique* of order n is a subset of n vertices of a graph such that each vertex is adjacent to every other vertex. An *independent set* is the complement of a clique. The maximum clique and maximum independent set of a graph G are the *clique number* $\omega(G)$ and the *independence number* $\alpha(G)$, respectively. A *cut* is a partition of the vertices of a graph into two sets, $S \subset V(G)$ and $\overline{S} = V(G) \setminus S$. The *size* of a cut is the number of edges that join the two parts, that is, $|\{\{u, v\} \in E(G) \mid u \in S \text{ and } v \in \overline{S}\}|$. MAX-CUT is a well-known combinatorial optimization problem that asks for the maximum size of a cut of a graph, which we denote as $MC(G)$. A set $D \subseteq V(G)$ is a *dominating set* of G if every vertex is either in D or adjacent to a vertex in D. A vertex u *dominates* vertex v if $u = v$ or $\{u, v\} \in E(G)$. The minimum order of such a set is the *domination number* and is denoted $\gamma(G)$.

An important concept of this thesis is that of the Ramsey arrowing operator. Given graphs $(G_1, G_2, \ldots, G_k)$, we write $G \to (G_1, G_2, \ldots, G_k)$ and say G *arrows* $(G_1, G_2, \ldots, G_k)$ if for any k-coloring of the edges of G, a monochromatic G_i exists for some color $i \in \{1, \ldots, k\}$. When $G_i = K_{s_i}$ for all i, we write $G \to (s_1, s_2, \ldots, s_k)$. The *Ramsey number* $R(G_1, G_2, \ldots, G_k)$ is

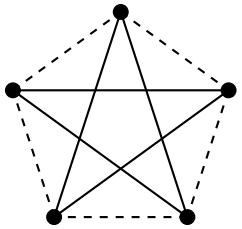


Figure 1.1: Unique coloring of K_5 showing $K_5 \not\to (3,3)$

the smallest n such that $K_n \to (G_1, G_2, \ldots, G_k)$. The *Folkman number* $F_e(s_1, s_2, \ldots, s_k; k)$ is the order of the smallest K_k-free graph that arrows $(s_1, s_2, \ldots, s_k)$.

We know that Ramsey numbers exist due to the seminal paper by Frank P. Ramsey in 1930 [125]. As previously mentioned, a classical small example is $R(3,3) = 6$. Note that this means that $K_5 \not\to (3,3)$ and $K_6 \to (3,3)$. The unique witness of $K_5 \not\to (3,3)$ is presented in Figure 1.1, where each of the two colors is isomorphic to C_5. Proving $K_6 \to (3,3)$ was a problem of the 1953 *William Lowell Putnam Mathematical Competition*. Consider a red-blue edge coloring of K_6. As $v \in V(G)$ is incident to 5 edges, at least three of them will be the same color, say red without loss of generality. The three other vertices incident to these red edges are also connected. None of these connections can be red, or else a red triangle is formed with v. However, if they are all colored blue, they form a blue triangle. Thus, $K_6 \to (3,3)$.

In general, research involving Ramsey numbers is split into two areas. In one, the question is how the numbers behave asymptotically, that is, how for example $R(3,k)$ behaves as k approaches infinity. The other is concerned with determining values and bounds for numbers with small parameters. A main goal of the latter is to provide insight that quantifies results of the former. In Chapter 3, we present what is known about the asymptotics of $R(C_4, K_m)$, and focus our work on computational attacks on the small numbers.

We approach these and related graph parameters and problems with a computational perspective. Determining $\omega(G)$, $\alpha(G)$, $MC(G)$, or $\gamma(G)$ for a general graph G is **NP**-hard, and the corresponding decision problems are **NP**-complete (see [54]). Many Ramsey graph coloring problems are **NP**-hard or lie even higher in the polynomial hierarchy; we discuss some such problems in Section 2.2. It is straightforward to see that deciding Ramsey arrowing is at least **NP**-complete, as $G \to (k,2)$ decides $\omega(G) \geq k$. The difficulty of determining

these properties is a main motivation for studying how computational techniques within combinatorics and optimization can aid in solving specific instances of them.

Linear Algebra

Through out this work, a variable in boldface represents a vector in either $\mathbb{R}^n$ or $\mathbb{Z}^n$, the vector spaces of all n-dimensional vectors with real or integer entries, respectively. The entries of vector $\mathbf{v}$ are denoted $v_1, v_2, \ldots, v_n$. Unless otherwise specified, $\|\mathbf{v}\|$ is the Euclidean norm of $\mathbf{v} \in \mathbb{R}^n$, defined as

$$\|\mathbf{v}\| = \sqrt{v_1^2 + v_2^2 + \cdots + v_n^2}.$$

We use $\mathbf{0}$ and $\mathbf{1}$ to denote vectors whose entries consist of all 0's or 1's, respectively. Matrices are represented with capital letters, such as, for example, A for the adjacency matrix of a graph, or V for a matrix with column vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k$.

Given a set of linear independent vectors $B = \{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n\}$, the *span* of B is the set of all linear combinations of them. Let $\text{span}(B) = S$; S is a *subspace* of $\mathbb{R}^n$ and B is a *basis* of S. If the linear combinations of B are restricted to those with integer coefficients, that is,

$$\text{span}_{\mathbb{Z}}(B) = \left\{ \sum_{i=1}^{n} x_i \mathbf{b}_i \mid x_i \in \mathbb{Z}, \text{ for } 1 \leq i \leq n \right\},$$

then $\mathcal{L} = \text{span}_{\mathbb{Z}}(B)$ is the *lattice* with basis B. Lattices are discussed in detail in Chapter 4.

Mathematical Programming

A *linear program* (LP) asks to find the optimal value of a linear function subject to linear equality and inequality constraints (see for example [140, 126]). The *standard form* of a linear program is the maximization of the function $f(\mathbf{x})$, $\mathbf{x} \in \mathbb{R}^n$, formulated as:

$$\begin{aligned} \text{Maximize} \quad & f(\mathbf{x}) = \sum_{i=1}^{n} c_i x_i \\ \text{subject to:} \quad & \sum_{i=1}^{n} a_{ij} x_i \leq b_j, \quad j = 1, 2, \ldots, m, \\ & x_i \geq 0, \quad i = 1, 2, \ldots, n. \end{aligned} \tag{1.1}$$

Note that (1.1) can be rewritten in matrix form as

$$\max\{\mathbf{c}^T\mathbf{x} \mid A\mathbf{x} \leq \mathbf{b} \text{ and } \mathbf{x} \geq \mathbf{0}\},$$

and that the minimization of $f(\mathbf{x})$ can be determined by the maximization of $-f(\mathbf{x})$.

Linear programming has many applications in a wide variety of areas, including business, economics, engineering, and operations research, and was first developed by Leonid Kantorovich in 1939 for use in World War II (see [80]). Many efficient algorithms exist for solving LPs, including the well-known *simplex* and *interior point* methods, the latter of which run in polynomial time. The efficiency of these algorithms relies on the geometry of the constraints (see [65, 143]). A set $S \subset \mathbb{R}^n$ is *convex* if and only if the line segment connecting any two $\mathbf{u}, \mathbf{v} \in S$, defined as $\{\alpha\mathbf{u} + (1-\alpha)\mathbf{v} \mid 0 \leq \alpha \leq 1\}$, is contained in S. If the constraints of an LP are viewed as a hyperplane in $\mathbb{R}^n$, then the intersection of their feasible regions forms a *polytope*, an n-dimensional geometric object with "flat" sides. The solution of the LP will lie on one of the points of this polytope, all of which form a convex set. The simplex method, for example, uses this geometry to "walk" from point to point until the optimal one is found.

An *integer program* (IP) is a mathematical optimization program that restricts some or all of its variables to integers. An *integer linear program* (ILP) is an LP with the additional restriction that $\mathbf{x} \in \mathbb{Z}^n$. Unlike linear programming, integer programming tends to be computationally difficult; ILP is known to be **NP**-hard. The additional restriction of $\mathbf{x} \in \{0,1\}^n$ is one of Karp's 21 **NP**-complete problems [83]. Although this implies that the existence of a polynomial-time algorithm for solving IPs is unlikely, the fact that many combinatorial optimization problems can be formulated as them has resulted the area becoming an extensive subject. Many techniques are known to perform well for certain problems (see e.g. [112, 81]). A main thrust of this thesis involves formulating the previously described hard graph problems as integer programs, and then using some heuristic or bounding technique to find or approximate a solution.

A *semidefinite program* (SDP) is a linear program where the solution vector $\mathbf{x}$ is replaced with a positive semidefinite matrix [55]. A matrix X is *positive semidefinite*, denoted $X \succeq 0$, if and only if $\mathbf{v}^T X\mathbf{v} \geq 0$ for all $\mathbf{v} \in \mathbb{R}^n$. The standard form of SDP is:

$$\begin{aligned} \text{Maximize} \quad & \sum_{i,j}^{n} c_{ij} x_{ij} \\ \text{subject to:} \quad & \sum_{i,j}^{n} a_{ijk} x_{ij} = b_k, \quad k = 1, 2, \ldots, m, \\ & X \succeq 0. \end{aligned} \tag{1.2}$$

SDPs are convex optimization problems and, similarly to linear programs, can be solved with efficient algorithms, such as interior point methods.

1.3 Computational Thrust

A substantial part of this thesis involved the development of software to generate and manipulate graphs for use in experiments. The base of our software includes a library containing a robust graph data structure; it was used in all three parts of this work. Vertices and their associated adjacency lists are represented as bitsets with basic set operations accomplished using bitwise operations, as described in [93]. Our library includes over 80 functions that perform tasks ranging from simple (such as adding edges and determining the minimum degree) to complex (such as computing MAX-CLIQUE with pruned backtracking). We also implemented functionality to generate a large variety of graphs, including random graphs, Paley graphs, and graphs joined from multiple smaller graphs. Such graphs are explained in more detail in Section 2.4.2. A notable tool we developed for the $F_e(3,3;4)$ research is `archer`, an interactive prompt that calls the library to create, manipulate, and output graphs in real-time.

The computational attack on $R(C_4, K_m)$ required us to implement a number of additional combinatorial algorithms optimized to be as fast and as cheap as possible. We routinely tested the software, and often modified the code multiple times a week during the experimentation phase. The software's success was partly due to our consideration of aspects often overlooked, such as minimizing the size of the data types we used. The computations were made possible due to the Open Science Grid (OSG), a multidisciplinary initiative joining the resources of various cyberinfrastructures to meet the needs of academic computing of all sizes. We performed an estimate of 200,000 CPU hours (22 years) of computation on the grid. Our

algorithms and use of the OSG is discussed in detail in Section 3.5.

LLL and related basis reduction algorithms were implemented with the use of FLENS [66], a C++ library that is essentially an intuitive wrapper for the established linear algebra libraries BLAS and LAPACK. We implemented the algorithms this way in order to make use of the graph libraries previously described. Additional functions were written to convert graph problems to search problems involving vectors of lattice bases. This is described in more detail in Section 4.4.6.

All code was written in C++ and most tests were performed on Linux systems. Various `bash` scripts were written for batch experiments. All source code can be found in public `github` repositories [97].

In addition to our own code, we made use of a number of third-party software packages. Graph isomorphism testing, an essential part of graph enumeration, was performed using Brendan McKay's well-known `nauty` software [109]. In some cases, our code was compiled directly with `nauty` libraries, while in others the standalone tools of the software were used. The work discussed in Chapter 2 makes use of a number of extra software, including MATLAB [108], SDP solvers [14, 72], and SAT solvers [56, 5]. The Number Theory Library by Victor Shoup [133] was called when operations under Galois fields were needed. Finally, `sage` [137], an open source mathematical software built on Python, was used for verification of properties of our data, special graph generation, and the analysis of some graph automorphism groups.

1.4 Structure of Thesis

The structure of this thesis is as follows:

Chapter 2 discusses edge Folkman problems concerning triangles. Specific focus is placed on the Folkman number $F_e(3,3;4)$, which asks for the smallest order of a K_4-free graph that is not the union of two triangle-free graphs. The main result of this work is an improvement of the upper bound to $F_e(3,3;4) \leq 786$. A significant aspect of this result is the use of the Goemans-Williamson MAX-CUT SDP relaxation.

Chapter 3 studies the Ramsey numbers $R(C_4, K_m)$, which is the smallest n such that every graph on n vertices contains either a C_4 or independent set of order m. We present known asymptotic results for these numbers as well as the values and bounds for small m.

We conclude the chapter with a discussion on the computational approach to attacking these numbers, and establish $R(C_4, K_9) = 30$ and $R(C_4, K_{10}) = 36$ with large grid computations on the Open Science Grid.

Chapter 4 includes an overview of the Lenstra-Lenstra-Lovász (LLL) algorithm, a well-known, polynomial-time algorithm that, when given a basis of a lattice, returns a *reduced* basis for the same lattice with the shortest vectors it can find. We include a summary of lattices and basis reduction techniques, the algorithm's applications to combinatorial computing, and present a method which makes use of it as a heuristic for computing a graph's domination number.

We present a number of theorems throughout this thesis. Those that do not contain citations are our original work, while those that include citations are previously known results. In some cases we provide proofs for the theorems which are not our own. This is mostly done to provide insight into our work, but sometimes proofs are presented simply because we found them interesting enough to do so.

Chapter 2

Folkman Number $F_e(3,3;4)$

2.1 Introduction

Given a graph G, we write $G \to (a_1, \ldots, a_k)$ and say that G *arrows* $(a_1, \ldots, a_k)$ if for every edge k-coloring of G, a monochromatic K_{a_i} is forced for some color $i \in \{1, \ldots, k\}$. Likewise, for graphs F and H, $G \to (F, H)$ if for every edge 2-coloring of G, a monochromatic F is forced in the first color or a monochromatic H is forced in the second. Define $\mathcal{F}_e(a_1, \ldots, a_k; p)$ to be the set of all graphs that arrow $(a_1, \ldots, a_k)$ and do not contain K_p; they are often called Folkman graphs. The edge Folkman number $F_e(a_1, \ldots, a_k; p)$ is the smallest order of a graph that is a member of $\mathcal{F}_e(a_1, \ldots, a_k; p)$. In 1970, Folkman [49] showed that for $k > \max\{s, t\}$, $F_e(s, t; k)$ exists. The related problem of vertex Folkman numbers $F_v(s, t; k)$, where vertices are colored instead of edges, is more studied (see e.g [106, 114]) than edge Folkman numbers, but we will not be discussing them in detail.

In 1967, Erdős and Hajnal [40] asked the question: Does there exist a K_4-free graph that is not the union of two triangle-free graphs? This question is equivalent to asking for the existence of a K_4-free graph such that in any edge 2-coloring, a monochromatic triangle is forced. After Folkman proved the existence of such a graph, the question then became to find how small this graph could be, or using the above notation, what is the value of $F_e(3, 3; 4)$. Prior to this work, the best known bounds for this number were $19 \leq F_e(3, 3; 4) \leq 941$ [124, 36].

An improvement to the upper bound of the Folkman number $F_e(s, t; k)$ requires one K_k-free witness that arrows (s, t), while an improvement to the lower bound requires a proof that all graphs of a given order have no such property. This is perhaps a reason for the puzzling large range between the lower and upper bounds of $F_e(3, 3; 4)$. Clearly in this case, both the upper and lower are difficult to improve.

2.1.1 Overview of $F_e(3,3;k)$

Table 2.1 summarizes known results for $F_e(3,3;k)$. Since the Ramsey number $R(3,3)=6$, it follows that $F_e(3,3;k)=6$ for $k \geq 7$. In 1968, Graham [61] responded to Erdős and Hajnal by presenting an explicit K_6-free graph on 8 vertices that arrows $(3,3)$. As no such graph exists with 7 vertices, this showed $F_e(3,3;6)=8$. This graph, $K_8 - C_5$, is displayed in Figure 2.1, and a summary of the proof that $K_8 - C_5 = K_3 + C_5 \to (3,3)$ is found in Theorem 1.

k	$F_e(3,3;k)$	Graphs	Who	Ref.
≥ 7	6	K_6	folklore	
6	8	$K_3 + C_5$	Graham 1968	[61]
5	15	659 graphs	Piwakowski et al. 1999	[120]
4	19 – 786	L_{786}	RX 2007, this work 2012	[124]

Table 2.1: Known values and bounds for $F_e(3,3;k)$

Theorem 1 (Graham, 1968 [61])**.** $G = K_8 - C_5 = K_3 + C_5 \to (3,3)$

Proof. Assume there is an edge coloring of G such that neither of the colors contain a triangle; call the parts of this coloring R (red) and B (blue). Consider the triangle of G that is joined to C_5. Two of the vertices in this triangle will be incident to a red and blue edge, as the K_3 is non-monochromatic. Let one of those vertices be v. This vertex will be adjacent to all five vertices $c_1, \ldots, c_5$ of the C_5. At least 3 of these edges will be one color, so without loss of generality, say $\{v,c_1\},\{v,c_2\},\{v,c_3\} \in B$. Two of $\{c_1,c_2,c_3\}$ must be adjacent, say $\{c_1,c_2\} \in E(C_5)$. Clearly, $\{c_1,c_2\}$ must be red to avoid a blue triangle. Pick $u \in V(K_3)$ such that $\{u,v\} \in B$. Then, neither $\{u,c_1\}$ nor $\{u,c_2\}$ can be in B or a blue triangle is formed. However, if they are both red, then they form a red triangle with $\{c_1,c_2\}$. Therefore, a monochromatic triangle must exist. ■

The case for $k=5$ received much attention up until 1999, when Piwakowski et al. determined that $F_e(3,3;5)=15$ [120]. The first upper bound $F_e(3,3;5) \leq 42$ was obtained by Schäuble in 1969 [131], although the proof of existence is credited to an unpublished work by Pósa. In 1971, Graham and Spencer [62] improved the bound to $F_e(3,3;5) \leq 23$. Both constructions rely on cleverly connecting a number of C_5 graphs and a triangle. The bound was then improved to 18 by Irving in 1973 [76], 16 by Hadziivanov and Nenov in

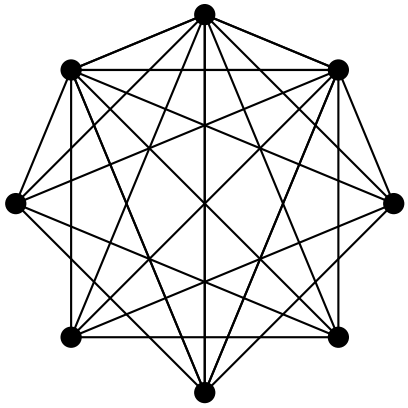


Figure 2.1: $K_8 - C_5$, the witness of $F_e(3,3;6) = 8$

1979 [67], and 15 by Nenov in 1981 [113]. The latter two results were published in Russian and seemed to go unnoticed for some time.

The computational approach by Piwakowski et al. to determine $F_e(3,3;5) \geq 15$ involved processing a large number of graphs to show that no 14-vertex graph exists in $\mathcal{F}_e(3,3;5)$. Since $R(3,5) = 14$, any 14-vertex graph $G \in \mathcal{F}_e(3,3;5)$ will contain a $\overline{K_3}$. They determined a number of properties of $G \setminus \overline{K_3}$, and all graphs on 11 vertices with these properties were processed in order to reconstruct graphs G. However, no such graphs were found.

Figure 2.2 presents the unique bicritical 15-vertex graph in $\mathcal{F}_e(3,3;5)$. It is bicritical because (a) adding any edge forms a K_5 and (b) removing any edge makes it not arrow $(3,3)$. This graph plays an important role in the vertex Folkman number $F_v(3,3;4)$, as removing vertex v yields the unique bicritical witness of $F_v(3,3;4) = 14$.

The focus of this chapter is on the most studied open Folkman number, $F_e(3,3;4)$, and ways the well-known graph MAX-CUT problem can determine arrowing of triangles. The next section overviews the rich history of this number.

2.2 History of $F_e(3,3;4)$

Table 2.2 summarizes the events surrounding $F_e(3,3;4)$, starting with Erdős and Hajnal's [40] original question of existence. After Folkman [49] proved the existence, Erdős, in 1975, offered \$100 for deciding if $F_e(3,3;4) < 10^{10}$.

Most work on the upper bound of $F_e(3,3;4)$ has made use of an idea originally presented

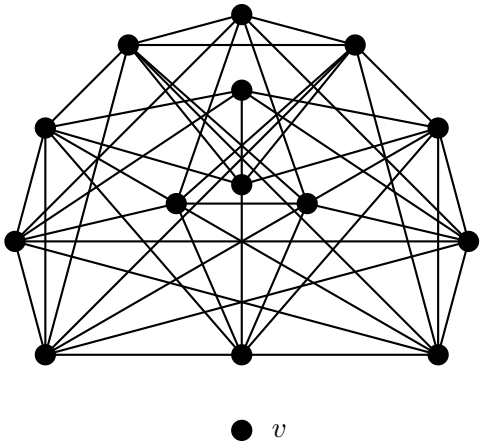


Figure 2.2: Only bicritical graph of all 659 witnesses to $F_e(3,3;5) = 15$, where v is connected to all other 14 vertices.

by Goodman in 1959 [60], which involves counting the triangles of an edge-colored graph. Note that there is essentially a single coloring of a non-monochromatic triangle: two edges are one color and one edge is the other. A non-monochromatic triangle therefore has two vertices that are incident to both a red and blue edge. Let G be an edge-colored graph with no monochromatic triangles. Let $t_{\mathrm{RB}}(x)$ count the triangles $\{x, y, z\}$ where $\{x, y\}$ is red and $\{x, z\}$ is blue; then, $\sum_{v \in V(G)} t_{\mathrm{RB}}(v) = 2t_{\triangle}(G)$. If G_x is the induced subgraph of $N_G(x)$, then each edge in G_x counts a triangle, yielding $\sum_{v \in V(G)} |E(G_v)| = 3t_{\triangle}(G)$. Since no monochromatic triangle exists, the vertices of each G_x can be partitioned such that $MC(G_x) = t_{\mathrm{RB}}(x)$. Combining these equations gives $\sum_{v \in V(G)} MC(G_v) = \frac{2}{3}|E(G_v)|$.

However, if every coloring of a graph G contains a monochromatic triangle, then some G_v can not be partitioned completely, resulting in Theorem 2.

Theorem 2 (Spencer, 1988 [135])**.** *Let G be a graph and G_v be the graph induced by $N_G(v)$. If*

$$\sum_{v \in V(G)} MC(G_v) < \frac{2}{3}|E(G_v)|,$$

then $G \to (3,3)$.

Deciding $F_e(3,3;4) < 10^{10}$ remained open for over 10 years. Frankl and Rödl [50] nearly met Erdős' request in 1986 when they showed that $F_e(3,3;4) < 7.02 \times 10^{11}$ using probabilistic arguments and ideas similar to those described above. In 1988, Spencer [135], in a seminal paper that also made use of probabilistic techniques, proved the existence of a Folkman

Year	Lower/Upper Bounds	Who/What	Ref.
1967	any?	Erdős-Hajnal	[40]
1970	exist	Folkman	[49]
1972	10 –	Lin	[102]
1975	– 10^{10}?	Erdős offers \$100 for proof	
1986	– 8×10^{11}	Frankl-Rödl	[50]
1988	– 3×10^9	Spencer	[135]
1992	– 10^6?	Erdős offers \$100 for proof	[19]
1999	16 –	Piwakowski et al. (implicit)	[120]
2007	19 –	Radziszowski-Xu	[124]
2008	– 9697	Lu	[105]
2008	– 941	Dudek-Rödl	[36]
2012	– 786	this work	
2012	– 100?	Graham offers \$100 for proof	

Table 2.2: Timeline of progress on $F_e(3,3;4)$.

graph of order 3×10^9 (after an erratum by Hovey) without explicitly constructing it. The main idea behind his result involved $G = G(n,p)$, the random graph with n vertices and edge probability p. From this graph, a K_4-free graph G^* is obtained by randomly removing an edge from each K_4 in G. By setting $n = 3 \times 10^9$, he showed that a G^* satisfying the condition in Theorem 2 exists with positive probability.

Erdős then offered \$100 for deciding if $F_e(3,3;4) < 10^6$ (see [19], page 46). Much time passed until 2007, when Lu and Dudek-Rödl independently showed it to be true. Lu determined $F_e(3,3;4) \leq 9697$ by constructing a family of K_4-free circulant graphs (which we discuss in Section 2.5) and showing that some such graphs arrow $(3,3)$ using a combination of spectral analysis and Theorem 2. The main idea behind his proof involves a graph H being δ-*fair* if $MC(H) < (\frac{1}{2} + \delta)|E(H)|$. From Theorem 2 it follows that if each H_v is $\frac{1}{6}$-fair, then $H \to (3,3)$. Lu was able to show that d-regular graphs were δ-fair if the smallest eigenvalue of the adjacency matrix was greater than $-2\delta d$, and found a number of "small" graphs, including one with order 9697, that had this property.

Dudek and Rödl reduced the upper bound to the best known to date, 941. Their method, which we have pursued further with some success, is discussed in the next section. A natural next question is whether $F_e(3,3;4) < 100$. During the 2012 SIAM Conference on Discrete

Mathematics in Halifax, Nova Scotia, Ronald Graham announced a $100 award for deciding this. We discuss a possible witness for this bound in Section 2.4.1.

The lower bound for $F_e(3,3;4)$ has been much less studied than the upper bound. Lin [102] obtained a lower bound on 10 in 1972. Without the help of a computer, he showed that $F_e(a_1,\ldots,a_k;R(a_1,\ldots,a_k)-1) \geq R(a_1,\ldots,a_k)+4$, giving $F_e(3,3;5) \geq 10$. The next improvement did not come until 1999 when $F_e(3,3;5) = 15$ [120] was determined. The 659 graphs on 15 vertices witnessing $F_e(3,3;5) = 15$ contain K_4, thus giving the bound $16 \leq F_e(3,3;4)$.

In 2007, Radziszowski and Xu gave a computer-free proof of $18 \leq F_e(3,3;4)$ and improved the lower bound further to 19 with the help of computations [124]. A summary of this work follows.

Theorem 3 (Radziszowski and Xu, 2007 [124])**.** $F_e(3,3;4) \geq 18$

Proof. To show that $F_e(3,3;4) \geq 18$, we must show that no K_4-free graph with 17 vertices arrows $(3,3)$. Define graph G_{17} as $V(G) = \mathbb{Z}_{17}$ and $E(G) = \{\{u,v\} \mid u - v = \alpha^2\}$, where $\alpha^2 \in \{1,2,4,8\}$. This circulant graph is the well-known Paley graph of order 17, has no K_4, and is the unique lower-bound witness to $R(4,4) = 18$ [46]. The subgraphs of G_{17} induced by distances $\{1,4\}$ and $\{2,8\}$ do not contain triangles, and therefore $G_{17} \notin \mathcal{F}_e(3,3;4)$. Assume there exists a graph $G \in \mathcal{F}_e(3,3;4)$; since G is non-isomorphic to G_{17} and does not contain a K_4, it must contain a $\overline{K_4}$. Connecting the vertices $\{v_1,v_2,v_3,v_4\}$ of this $\overline{K_4}$ with the other 13 vertices of G_{17} does not cause a K_5, and thus the resulting graph G' is in $\mathcal{F}_e(3,3;5)$. However, since the edges incident to the $\overline{K_4}$ do not form a triangle with each other, $G' \setminus \{v_1,v_2,v_3\}$ is also in $\mathcal{F}_e(3,3;5)$. This contradicts $F_e(3,3;5) = 15$ and thus $F_e(3,3;4) \geq 18$. ■

The proof of $F_e(3,3;4) \geq 19$ follows the same general idea, but is slightly more complicated due to a larger number of graphs involved. If a 18-vertex graph $G \in \mathcal{F}_e(3,3;4)$ exists, then because $R(4,4) = 18$, it must contain a $\overline{K_4}$. Radziszowski and Xu showed that $G \setminus \overline{K_4}$ must be isomorphic to one of the 153 14-vertex graphs in $\mathcal{F}_v(3,3;4)$. They then used computations to process all 153 such graphs and reconstruct the possible graphs G. However, all graphs

reconstructed did not arrow $(3,3)$, showing $F_e(3,3;4) > 18$.

The long history of $F_e(3,3;4)$ is not only interesting in itself but also provides insight into how difficult the problem is. Finding good bounds on the smallest order of any Folkman graph (with fixed parameters) seems to be difficult, and some related Ramsey graph coloring problems are **NP**-hard or lie even higher in the polynomial hierarchy. For example, Burr showed that arrowing $(3,3)$ is **coNP**-complete (see [54]), and Schaefer [130] showed that for general graphs F, G, and H, $F \to (G,H)$ is $\mathbf{\Pi_2^P}$-complete. The latter result is particularly significant, as it provides a natural problem that is complete for a higher level of the polynomial hierarchy.

2.3 Arrowing and MAX-CUT

Building off Spencer's and the other methods described above, Dudek and Rödl [36] in 2008 showed how to construct a graph H_G from a graph G, such that the maximum size of a cut of H_G determines whether or not $G \to (3,3)$. They construct the graph H_G as follows. The vertices of H_G are the edges of G, so $|V(H_G)| = |E(G)|$. For $e_1, e_2 \in V(H_G)$, if edges $\{e_1, e_2, e_3\}$ form a triangle in G, then $\{e_1, e_2\}$ is an edge in H_G.

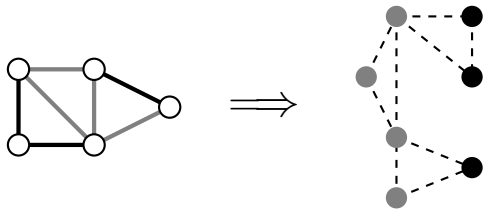


Figure 2.3: Converting G to H_G

Let $t_\triangle(G)$ denote the number of triangles in graph G. Clearly, $|E(H_G)| = 3t_\triangle(G)$. Let $MC(H)$ denote the MAX-CUT size of graph H.

Theorem 4 (Dudek and Rödl, 2008 [36])**.** *$G \to (3,3)$ if and only if $MC(H_G) < 2t_\triangle(G)$.*

There is a clear intuition behind Theorem 4 that we will now describe. Any edge 2-coloring of G corresponds to a bipartition of the vertices in H_G. If a triangle colored in G is not monochromatic, then its three edges, which are vertices of H_G, will be separated in the bipartition. If we treat this bipartition as a cut, then the size of the cut will count each

triangle twice for the two edges that cross it. Since there is only one triangle in a graph that contains two given edges, this effectively counts the number of non-monochromatic triangles. Therefore, if it is possible to find a cut that has size equal to $2t_{\triangle}(G)$, then such a cut defines an edge coloring of G that has no monochromatic triangles. However, if $MC(H_G) < 2t_{\triangle}(G)$, then in each coloring, all three edges of some triangle are in one part and thus, $G \to (3,3)$.

A benefit of converting the problem of arrowing $(3,3)$ to MAX-CUT is that the latter is well-known and has been studied extensively in computer science and mathematics (see for example [30]). The decision problem MAX-CUT(H,k) asks whether or not $MC(H) \geq k$. It is known that MAX-CUT is **NP**-hard and the decision version was one of Karp's 21 **NP**-complete problems [83]. In our case, $G \to (3,3)$ if and only if MAX-CUT$(H_G, 2t_{\triangle}(G))$ doesn't hold. Since MAX-CUT is **NP**-hard, an attempt is often made to approximate it, such as in the approaches presented in the next two sections.

2.3.1 Minimum Eigenvalue Method

A method exploiting eigenvalues was used by Dudek and Rödl [36] to show that some large graphs are members of $\mathcal{F}_e(3,3;4)$. The following upper bound (2.1) on $MC(H_G)$ can be found in [36], where $\lambda_{\min}$ denotes the minimum eigenvalue of the adjacency matrix of H_G.

$$MC(H_G) \leq \frac{|E(H_G)|}{2} - \frac{\lambda_{\min}|V(H_G)|}{4}. \tag{2.1}$$

The proof of this bound is quite simple. Let $\mathbf{x} = [x_1, x_2, \ldots, x_n]$ and $x_i \in \{-1, 1\}$ for all $1 \leq i \leq n$. For a cut $\{S, \overline{S}\}$ of $H_G = (V, E)$, let $x_i = 1$ if vertex i is in S and $x_i = -1$ if i is in $\overline{S}$. Clearly, $\frac{1}{4}\sum_{\{i,j\}\in E}(x_i - x_j)^2$ counts the size of the cut. Let A be the adjacency matrix of H_G, where $a_{ij} = 1$ if $\{i,j\} \in E(H_G)$ and $a_{ij} = 0$ otherwise. Then,

$$\begin{aligned}
\sum_{\{i,j\}\in E}(x_i - x_j)^2 &= \sum_{\{i,j\}\in E} x_i^2 + x_j^2 - 2x_ix_j \\
&= \sum_{i=1}^{n} \deg(i) \cdot x_i^2 - \sum_{\{i,j\}\in E} 2x_ix_j \\
&= \sum_{i=1}^{n} \deg(i) - \sum_{i,j} a_{ij}x_ix_j \\
&= 2|E| - \mathbf{x}^T A\mathbf{x}.
\end{aligned}$$

Because A is symmetric, from the Rayleigh-Ritz ratio (see e.g). Theorem 4.2.2 in [74]), we know that $\mathbf{y}^T A\mathbf{y} \geq \lambda_{\min}\|\mathbf{y}\|^2$ for all $\mathbf{y} \in \mathbb{R}^n$. Then, $2|E| - \mathbf{x}^T A\mathbf{x} \leq 2|E| - \lambda_{\min}\|\mathbf{x}\|^2$ and $\|\mathbf{x}\|^2 = |V|$, giving the inequality in (2.1).

Dudek and Rödl used (2.1) to prove the following theorem:

Theorem 5 (Dudek and Rödl, 2008 [36])**.** $F_e(3,3;4) \leq 941$

Proof. For positive integers r and n, if -1 is an r-th residue modulo n, then let $G(n,r)$ be a circulant graph on n vertices with the vertex set $\mathbb{Z}_n$ and the edge set $E(G(n,r)) = \{\{u,v\} \mid u \neq v \text{ and } u - v \equiv \alpha^r \bmod n, \text{ for some } \alpha \in \mathbb{Z}_n\}$.

The graph $G_{941} = G(941,5)$ has 707632 triangles. Using the MATLAB [108] `eigs` function, Dudek and Rödl [36] computed

$$MC(H_{G_{941}}) \leq 1397484 < 1415264 = 2t_{\triangle}(G_{941}).$$

Thus, by Theorem 1, $G_{941} \to (3,3)$. ■

In an attempt to improve $F_e(3,3;4) \leq 941$, we removed vertices of G_{941} to see if the minimum eigenvalue bound would still show arrowing. We applied multiple strategies for removing vertices, including removing neighborhoods of vertices, randomly selected vertices, and independent sets of vertices. Most of these strategies were successful, and led to the following theorem:

Theorem 6. $F_e(3,3;4) \leq 860$.

Proof. For a graph G with vertices $\mathbb{Z}_n$, define $C = C(d,k) = \{v \in V(G) \mid v = id \bmod n, \text{ for } 0 \leq i < k\}$. Let $G = G_{941}$, $d = 2$, $k = 81$, and G_C be the graph induced on $V(G) \setminus C(d,k)$. Then G_C has 860 vertices, 73981 edges and 542514 triangles. Using the MATLAB `eigs` function, we obtain $\lambda_{\min} \approx -14.663012$. Setting $\lambda_{\min} > -14.664$ in (2.1) gives

$$MC(H_{G_C}) < 1084985 < 1085028 = 2t_{\triangle}(G_C). \tag{2.2}$$

Therefore, $G_C \to (3,3)$. ■

None of the methods used allowed for 82 or more vertices to be removed without the upper bound on MC becoming larger than $2t_\triangle$.

Small Examples

Although the minimum eigenvalue method led to the above results, it does not always show arrowing for small known examples. Let α be the upper bound of $MC(H_G)$ computed with this method and let $\beta = 2t_\triangle(G)$.

When $G = K_6$, the upper bound witness for $R(3,3) = 6$, this method does work. We construct H_{K_6} and obtain $|V(H_{K_6})| = 15$ and $|E(H_{K_6})| = 3t_\triangle(K_6) = 60$. We compute $\lambda_{\min}(H_{K_6}) = -2$ and

$$\alpha = \frac{60}{2} - \frac{(-2)(15)}{4} = 37.5, \qquad \beta = 40.$$

Since $\alpha < \beta$, the $\lambda_{\min}$ method successful shows that $K_6 \to (3,3)$.

However, the method fails for the next simplest case, $G = K_3 + C_5$. We construct H_G, with $|V(H)| = 23$ and $|E(H)| = 93$, and compute $\lambda_{\mathsf{min}}(H_G) \approx -3.3393$. Then,

$$\alpha = \frac{93}{2} - \frac{(-3.3393)(23)}{4} = 65.6993, \qquad \beta = 62.$$

Since $\alpha > \beta$, we cannot determine $K_3 + C_5 \to (3,3)$ using this method. The fact that (2.1) fails for this case was a main motivation for finding other methods which place upper bounds on the MAX-CUT of a graph. The next section discusses the *Goemans-Williamson semi-definite programming MAX-CUT relaxation*, which we used successfully to further improve the upper bound of $F_e(3,3;4)$.

2.3.2 Goemans-Williamson Method

The Goemans-Williamson MAX-CUT approximation algorithm [58] is a well-known, polynomial-time algorithm that relaxes the problem to a semidefinite program (SDP). It involves the first use of SDP in combinatorial approximation and has since inspired a variety of other successful algorithms (see e.g. [82, 51]). This randomized algorithm returns a cut with

expected size at least 0.87856 of the optimal value. However, in our case, all that is needed is a feasible solution to the SDP, as it gives an upper bound on $MC(H)$. A brief description of the Goemans-Williamson relaxation follows.

The first step in relaxing MAX-CUT is to represent the problem as a quadratic integer program. Given a graph H with $V(H) = \{1, \ldots, n\}$ and nonnegative weights $w_{i,j}$ for each pair of vertices $\{i, j\}$, we can write the MAX-CUT of H as the following objective function:

$$\begin{aligned} \text{Maximize} \quad & \frac{1}{2} \sum_{i<j} w_{i,j}(1 - y_i y_j) \\ \text{subject to:} \quad & y_i \in \{-1, 1\} \quad \text{for all } i \in V(H). \end{aligned} \tag{2.3}$$

Define one part of the cut as $S = \{i \mid y_i = 1\}$. Since in our case all graphs are weightless, we will use

$$w_{i,j} = \begin{cases} 1 & \text{if } \{i, j\} \in E(H), \\ 0 & \text{otherwise.} \end{cases}$$

Next, the integer program (2.3) is relaxed by extending the problem to higher dimensions. Each $y_i \in \{-1, 1\}$ is now replaced with a vector on the unit sphere $\mathbf{v}_i \in \mathbb{R}^n$, as follows:

$$\begin{aligned} \text{Maximize} \quad & \frac{1}{2} \sum_{i<j} w_{i,j}(1 - \mathbf{v}_i \cdot \mathbf{v}_j) \\ \text{subject to:} \quad & \|\mathbf{v}_i\| = 1 \quad \text{for all } i \in V(H). \end{aligned} \tag{2.4}$$

If we define a matrix Y with the entries $y_{i,j} = \mathbf{v_i} \cdot \mathbf{v_j}$, that is, the Gram matrix of $\mathbf{v}_1, \ldots, \mathbf{v}_n$, then $y_{i,i} = 1$ and Y is positive semi-definite. Therefore, (2.4) is a semidefinite program. We can write the SDP in the same form as (1.2).

$$\begin{aligned} \text{Maximize} \quad & \frac{1}{2} \sum_{i<j} w_{i,j}(1 - y_{i,j}) \\ \text{subject to:} \quad & y_{i,i} = 1 \quad \text{for all } i \in V(H), \\ & Y \succeq 0. \end{aligned} \tag{2.5}$$

Cholesky decomposition can then be performed on Y to obtain the vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$. Then,

a simple rounding technique is used to obtain an approximate cut. The main idea is to generate a random uniformally distributed vector $\mathbf{r}$ and let $S^* = \{i \mid \mathbf{v}_i \cdot \mathbf{r} \geq 0\}$ be one part of the cut. The vector $\mathbf{r}$ is interpreted as the normal of a hyperplane that "cuts" the unit sphere, partitioning the unit vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$ into two parts. If $MC^*(H)$ is the size of the cut $\{S^*, \overline{S^*}\}$, then some analysis yields $\mathrm{E}[MC^*(H)] \geq \alpha_{\mathrm{GW}} MC(H)$, where $\mathrm{E}[MC^*(H)]$ is the expected value and $\alpha_{\mathrm{GW}} > 0.87856$. However, as the actual maximum value of (2.5) is an upper bound on $MC(H)$, completing this last step is out of the scope of this work.

2.4 Experiments

Using the Minimum Eigenvalue and Goemans-Williamson approaches, we tested a wide variety of graphs for arrowing by finding upper bounds on MAX-CUT. These graphs included the $G(n, r)$ graphs tested by Dudek and Rödl, similar circulant graphs based on the Galois fields $GF(p^k)$, and different types of random graphs. Various modifications of these graphs were also considered, including the removal and/or addition of vertices and/or edges, as well as copying or joining multiple candidate graphs together in various ways. We detail such experiments in this section.

Multiple SDP solvers that were designed [14, 72] to handle large-scale SDP and MAX-CUT problems were used for the tests. Specifically, we made use of a version of `SDPLR` by Samuel Burer [14], a solver that uses low-rank factorization. This version, `SDPLR-MC`, includes specialized code for the MAX-CUT SDP relaxation. `SBmethod` by Christoph Helmberg [72] implements a spectral bundle method and was also applied successfully in our experiments. In all cases where more than one solver was used, the same results were obtained.

Throughout this section, we use α to denote the computed upper bound of $MC(H_G)$ and β to denote $2t_{\triangle}(G)$. We make use of the parameter $\rho = (\alpha - \beta)/\alpha$, as defined by Dudek and Rödl [36], to estimate how "close" the methods are to showing $G \to (3, 3)$.

2.4.1 Graphs

We tested the graph G_C of Theorem 6 with the SDP relaxation and obtained the upper bound $MC(H_{G_C}) \leq 1077834$, a significant improvement over the bound 1084985 obtained from the minimum eigenvalue method. This provides additional proof that $G_C \to (3, 3)$, and is an example of when (2.4) yields a much better upper bound.

The type of graph that led to the best results, including an improvement to the upper bound of $F_e(3,3;4)$, was described by Lu in [105]. We discuss these graphs and our results in the next section.

Graph G_{127}

Define graph G_{127} as $V(G_{127}) = \mathbb{Z}_{127}$ and $E(G_{127}) = \{\{x,y\} \mid x - y \equiv \alpha^3 \bmod 127\}$ (that is, the graph $G(127,3)$ as defined in Section 2.3.1). We have given this graph particular attention, as it has been conjectured by Exoo that $G_{127} \to (3,3)$. He also suggested that subgraphs induced by less than 100 vertices of G_{127} may as well, which would give a positive answer to Graham's question of whether $F_e(3,3;4) < 100$.

G_{127} has 2667 edges, 9779 triangles, is K_4-free, and has an independence number of 11. It is regular of degree 42 and is both vertex- and edge-transitive. The graph was originally defined by Hill and Irving in 1982 [73] and was used to show $R(4,4,4) \geq 128$, as the edges of K_{127} can be three-colored in such a way that each color is isomorphic to it.

An upper bound of 20181 for $MC(H_{G_{127}})$ was obtained by both the $\lambda_{\min}$ and SDP methods. As $2t_{\triangle}(G_{127}) = 19558$, the approaches fail to show $G_{127} \to (3,3)$. However, the "closeness" obtained is $\rho = 0.03088$, a relatively low value. Multiple attempts were made at modifying G_{127} in order to lower ρ, including removing edges and vertices, and multiple copies of G_{127} were attached together in a variety of ways. However, in every case, the modified graph had a ρ value greater than 0.03088.

G_{127} contains three disjoint independent sets of order 11. These sets were removed one-by-one and the resulting graphs were tested for arrowing. The results are presented in Table 2.3. Note that although ρ increases for both methods, the SDP ρ increases much less. This was a common trend among all experiments performed; the SDP bounds tended to be better, and was especially so when the graph had less symmetrical structure.x

# Removed	$\lvert E(G)\rvert$	$2t_{\triangle}(G)$	$\lambda_{\min}$	$\rho(\lambda_{\min})$	SDP	ρ(SDP)
0	2667	19558	20181	0.03088	20181	0.03088
1	2205	14476	15285	0.05293	15073	0.03961
2	1801	10670	11529	0.07451	11213	0.04843
3	1455	7836	8617	0.09064	8307	0.05670

Table 2.3: MAX-CUT tests with G_{127} and its independent sets removed

Circulant Graphs

A number of circulant graphs not defined by residues were tested. One such graph G_{199} was given particular attention, as it appears to be a viable candidate for arrowing $(3,3)$. G_{199} is defined as $V(G_{199}) = \mathbb{Z}_{199}$ and $E(G_{199}) = \{\{u,v\} \mid u - v \in D\}$, where

```
D = { 1, 2,  4, 13, 15, 19, 21, 24, 26, 27, 30, 33, 37, 38,
        42, 43, 48, 51, 58, 74, 76, 83, 84, 86, 92, 93, 96  }.
```

G_{199} is 54-regular with 5373 edges and 21492 triangles, and does not contain a K_4. The $\lambda_{\min}$ method gave $\alpha = 45497$ and $\rho = 0.05523$, while the SDP method gave $\alpha = 45173$ and $\rho = 0.04846$. Although these tests failed to show $G_{199} \to (3,3)$, the ρ values are still relatively low, and it is still quite possible that $G_{199} \in \mathcal{F}_e(3,3;4)$.

Additional Graphs

The $G(n,r)$ graphs given by Dudek-Rödl were tested for all primes $100 \leq n \leq 941$ and all our results agreed with theirs. Similar residue-based circulant graphs with prime-power orders, built over Galois fields, were also tested. Generating such graphs was accomplished with the Number Theory Library by Victor Shoup [133], a C++ library that includes data structures and algorithms for performing operations on polynomials over finite fields. Unfortunately, most graphs generated this way contained many K_4's, and those that did not performed poorly with the MAX-CUT tests.

Numerous types of random graphs were tested. Graphs $G(n,p)$ with varying $50 \leq n \leq 1000$ and p were made K_4-free by removing a random edge from each K_4. Graphs were also generated by randomly permuting all possible edges, and adding them via the random order when no K_4 was formed. Circulant graphs were generated in a similar way: for a graph on n vertices, the possible distances $1, 2, \ldots, \lfloor n/2 \rfloor$ were randomly permuted and the circulant edges were added in this order if no K_4 was formed. No such graphs generated by any of these approaches were feasible Folkman candidates, and both MAX-CUT methods failed significantly, with ρ values often in the range $(0.1, 0.4)$. This possibly suggests that well-structured graphs such as G_{127} and G_{199} are more likely to arrow $(3,3)$, and are better suited for such testing.

2.4.2 SAT-solvers

In addition to the MAX-CUT methods, testing of graphs was done using a reduction from arrowing triangles to the Boolean satisfiability problem, 3SAT. An instance of 3SAT consists of a Boolean formula in *conjunctive normal form*, that is, a conjunction of clauses where each clause is a disjunction of, in this case, three literals. The goal is to decide whether the formula can be *satisfied* (evaluated to TRUE) by some assignment of the variables. The general SAT problem was the first known **NP**-complete problem as shown in the well-known Cook-Levin Theorem [32, 101].

Given graph G, we can decide $G \to (3,3)$ by deciding the satisfiability of the Boolean formula $\phi(G)$ (see e.g. [124]), constructed as follows. For all $e_1, e_2, e_3 \in E(G)$ such that $\{e_1, e_2, e_3\}$ is a triangle, we add the clauses $(e_1 \vee e_2 \vee e_3)$ and $(\overline{e_1} \vee \overline{e_2} \vee \overline{e_3})$ to $\phi(G)$. Then,

$$G \not\to (3,3) \quad \text{iff} \quad \phi(G) \text{ is satisfiable.}$$

The assignments of TRUE and FALSE to the literals are equivalent to the assignments of *red* and *blue* to the edges. The pair of clauses corresponding to a triangle $\{e_1, e_2, e_3\}$ evaluates to TRUE only when the triangle is non-monochromatic, as an edge assigned TRUE yields $(e_1 \vee e_2 \vee e_3)$ TRUE and an edge assigned FALSE yields $(\overline{e_1} \vee \overline{e_2} \vee \overline{e_3})$ TRUE. Thus, $\phi(G)$ is satisfied only when every triangle is non-monochromatic.

Large 3SAT instances can often be solved using specialized software, most of which compete in the biennial international SAT competition [31]. We used a number of these SAT-solvers for additional testing of Folkman graph candidates. The software included `clasp` [56], which won one silver and two gold medals in the "Crafted" 2009 competition, and `glucose` [5], which won a gold medal in the "Application" 2011 competition.

Unfortunately, the SAT-solvers were unable to determine any cases of arrowing which were not previously known, or determined by MAX-CUT.

2.5 $F_e(3,3;4) \leq 786$

In this section, we discuss a set of graphs that Lu [105] used to show $F_e(3,3;4) \leq 9697$. We obtain a new upper bound on $F_e(3,3;4)$ using a modification of one such graph.

For positive integers n and s, $s < n$, s relatively prime to n, define set $S = \{s^i \bmod n \mid i =$

$0, 1, \ldots, m-1\}$, where m is the smallest positive integer such that $s^m \equiv 1 \bmod n$. If $-1 \bmod n \in S$, then let $L(n, s)$ be a circulant graph on n vertices with $V(L(n, s)) = \mathbb{Z}_n$. For vertices u and v, $\{u, v\}$ is an edge of $L(n, s)$ if and only if $u - v \in S$. Note that the condition that $-1 \bmod n \in S$ implies that if $u - v \in S$ then $v - u \in S$.

In Table 1 of [105], a set of potential members of $\mathcal{F}_e(3, 3; 4)$ of the form $L(n, s)$ were listed, and the graph $L(9697, 4)$ was shown to arrow $(3, 3)$. Lu gave credit to Exoo for showing that $L(17, 2)$, $L(61, 8)$, $L(79, 12)$, $L(421, 7)$, and $L(631, 24)$ do not arrow $(3, 3)$.

We tested all graphs from Table 1 of [105] of order less than 941 with the MAX-CUT method, using both the minimum eigenvalue and SDP upper bounds. Table 2.4 lists the results. Note that although none of the computed upper bounds of the $L(n, s)$ graphs imply arrowing $(3, 3)$, all SDP bounds match those of the minimum eigenvalue bound. This is distinct from other families of graphs, including those in [36], as the SDP bound is usually tighter. Thus, these graphs were given further consideration.

G	$2t_{\triangle}(G)$	$\lambda_{\min}$	SDP
$L(127, 5)$	19558	20181	20181
$L(457, 6)$	347320	358204	358204
$L(761, 3)$	694032	731858	731858
$L(785, 53)$	857220	857220	857220
G_{786}	857762	857843	857753

Table 2.4: Potential $\mathcal{F}_e(3, 3; 4)$ graphs G and upper bounds on $MC(H_G)$, where "$\lambda_{\min}$" is the bound (2.1) and "SDP" is the solution of (2.4) from `SDPLR-MC` and `SBmethod`. G_{786} is the graph of Theorem 7.

Numerous attempts were made at modifying these graphs in hopes that one of the MAX-CUT methods would be able to prove arrowing. $L(127, 5)$ was given particular attention, as it is the same graph as G_{127} discussed in the previous section. Although we were unable to obtain results with $L(127, 5)$, we were able to do so with $L(785, 53)$. Notice that all of the upper bounds for $MC(H_{L(785,53)})$ are 857220, the same as $2t_{\triangle}(L(785, 53))$. Our goal was then to slightly modify $L(785, 53)$ so that this value becomes smaller. Let G_{786} denote the

graph $L(785, 53)$ with one additional vertex v connected to the following 60 vertices:

```
{  0,   1,   3,   4,   6,   7,   9,  10,  12,  13,  15,  16,
  18,  19,  21,  22,  24,  25,  27,  28,  30,  31,  33,  34,
  36,  37,  39,  40,  42,  43,  45,  46,  48,  49,  51,  52,
  54,  55,  57,  58,  60,  61,  63,  66,  69, 201, 204, 207,
 210, 213, 216, 219, 222, 225, 416, 419, 422, 630, 642, 645  }
```

These vertices were found with a simple greedy approach: v was connected to vertices 0 to 784 in order if no K_4 was formed.

G_{786} is still K_4-free, has 61290 edges, and has 428881 triangles. The upper bound computed from the SDP solvers for $MC(H_{G_{786}})$ is 857753. We did not find a nice description for the vectors of this solution. Software implementing `SpeeDP` by Grippo et al. [63], an algorithm designed to solve large MAX-CUT SDP relaxations, was used by Rinaldi (one of the authors of [63]) to analyze this graph. He was able to obtain the bounds $857742 \leq MC(H_{G_{786}}) \leq 857750$, which agrees with, and improves over our upper bound computation. Since $2t_{\triangle}(G_{786}) = 857762$, we have both from our tests and his `SpeeDP` test that $G_{786} \to (3, 3)$, and the following main result.

Theorem 7. $F_e(3, 3; 4) \leq 786$.

We note that finding a lower bound on MAX-CUT, such as the $857742 \leq MC(H_{G_{786}})$ bound from `SpeeDP`, follows from finding an actual cut of a certain size. This method may be useful, as finding a cut of size $2t_{\triangle}(G)$ shows that $G \not\to (3, 3)$.

2.6 Concluding Remarks

Improving the upper bound of 786 is the main challenge involved with $F_e(3, 3; 4)$. The question of whether $G_{127} \to (3, 3)$ is still open, and any method that could solve it would be of much interest, as it would most likely aid in deciding whether $F_e(3, 3; 4) < 100$.

Our experiments have suggested that the SDP MAX-CUT relaxation of Goemans and Williamson produces tighter upper bounds on $MC(H)$ than those of the minimum eigenvalue method. This is especially apparent when H has less symmetrical structure. However, both methods appear insufficient for further improvements to the upper bound. They both fail to show arrowing for easy cases, such as all 659 15-vertex graphs in $\mathcal{F}_e(3, 3; 5)$ and some other small cases presented in Table 2.5. A possible reason for these failures is that the small

G	$\lambda_{\min}$	SDP
K_6	Pass	Pass
$K_3 + C_5$	Fail	Fail
$K_4 + C_5$	Fail	Pass

Table 2.5: Inconsistent results from MAX-CUT arrowing tests.

order of the graphs leaves little room for the error inherent in approximations, suggesting that the approximations work well when the graphs are sufficiently large. This seems to create a gap, where graphs of interest such as G_{127} are too small for approximation methods like SDP-solvers but are too large for exact methods like SAT-solvers.

It is therefore likely that a new method is needed for further improvements. A possible strategy is to attempt the computation of the exact solution of the MAX-CUT IP (2.3) via approaches like Rendl, Rinaldi, and Wiegele's SDP based branch & bound algorithm [128] used in their Biq Mac software [127]. Another possible thread of work is to attempt to prove $\phi(G)$ is unsatisfiable with methods different than exact SAT-solvers. For example, computing an upper bound on the maximum number of satisfiable clauses can potentially show unsatisfiability. Approximation algorithms for MAX-SAT, such as Karloff and Zwick's SDP based algorithm [82] and Maaren, Norden, and Heule's sums of squares based algorithm [139], may be worthy of investigation.

Another open question is the lower bound on $F_e(3,3;4)$, as it is quite puzzling that only 19 is the best known. Even an improvement to $20 \leq F_e(3,3;4)$ would be good progress.

Chapter 3

Ramsey Numbers $R(C_4, K_m)$

3.1 Introduction

Let G and H be simple graphs. An n-vertex graph F is a $(G, H; n)$-graph if it contains no subgraph isomorphic to G and $\overline{F}$ contains no subgraph isomorphic to H. Define $\mathcal{R}(G, H; n)$ to be the set of all such graphs. The Ramsey number $R(G, H)$ is the smallest n such that for every two-coloring of the edges of K_n, a monochromatic copy of G or H exists in the first or second color, respectively. Clearly, if a $(G, H; n)$-graph exists, then $R(G, H) > n$. It is known that Ramsey numbers exist [125] for all G and H. The values and bounds for various types of such numbers are collected and regularly updated by Radziszowski [121].

The cycle-complete Ramsey numbers $R(C_n, K_m)$ have received much attention, both theoretically and computationally. For fixed $n = 3$, the numbers are $R(3, k)$, one of the most studied Ramsey numbers (see [136]). Since 1976, it has been conjectured that $R(C_n, K_m) = (n-1)(m-1)+1$ for all $n \geq m \geq 3$, except $n = m = 3$ [48, 39]. Note that the lower bound is easy: $(m-1)$ vertex-disjoint copies of K_{n-1} provides a witness for $R(C_n, K_m) > (n-1)(m-1)$. For over 30 years, much work has been done to verify the upper bound, with $m = 8$ being the current smallest open case. Table 3.1 presents all known values and bounds for small $R(C_n, K_m)$.

This work involves fixed $n = 4$, that is, the case of avoiding the quadrilateral C_4 in the first color. Possibly the most puzzling aspect of these numbers is that exact asymptotics are unknown, unlike those for $R(C_3, K_m)$ and related multicolored Ramsey numbers. Considering that these numbers are a natural next step from $R(C_3, K_k) = R(3, k)$ makes them of particular interest. This work focuses on values and bounds of $R(C_4, K_m)$ for small m, and the computational methods involved in enumerating $(C_4, K_m; n)$-graphs. Figure 3.1 displays a $(C_4, K_4; 9)$-graph, a lower bound witness to $R(C_4, K_{10}) = 10$. Prior to this work,

K_n \ C_m	C_3	C_4	C_5	C_6	C_7	C_8	C_9
K_3	6	7	9	11	13	15	17
K_4	9	10	13	16	19	22	25
K_5	14	14	17	21	25	29	33
K_6	18	18	21	26	31	36	41
K_7	23	22	25	31	37	43	49
K_8	28	26	29-33	36	43	50	57
K_9	36	30∗					65?
K_{10}	40-42	36∗					

Table 3.1: $R(C_m, K_n)$ for small m and n (this work $*$). For references, see [121].

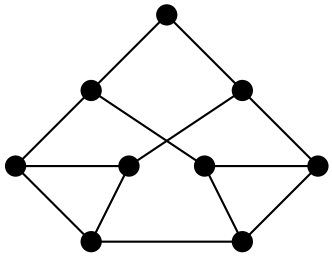


Figure 3.1: A $(C_4, K_4; 9)$-graph

the exact values for $R(C_4, K_m)$ were known for $3 \leq m \leq 8$. In this chapter, we present a computational proof that $R(C_4, K_9) = 30$ and $R(C_4, K_{10}) = 36$.

This chapter is outlined as follows. We first discuss the known asymptotics bounds of $R(C_n, K_m)$ and possible strategies for their improvement. Then, we discuss related topics in extremal graph theory that involve C_4-free graphs, as many such results are useful in the study of these Ramsey numbers. We conclude with descriptions of methods for proving values of small numbers, including our new computational results.

3.2 Asymptotics

The current best known asymptotic bounds for $R(C_4, K_m)$ are,

$$c_1 \left(\frac{n}{\log n}\right)^{\frac{3}{2}} \leq R(C_4, K_n) \leq c_2 \left(\frac{n}{\log n}\right)^2, \tag{3.1}$$

where c_1 and c_2 are positive constants.

The lower bound was obtained by Spencer in 1977 [134] using the well-known probabilistic

method, which we briefly discussed in Section 2.2. The upper bound was published by Caro, Li, Rousseau, and Zhang in 2000 [16], who in turn gave credit to an unpublished work by Szemerédi. The main challenge is determining whether $R(C_4, K_n) < n^{2-\epsilon}$ for some $\epsilon > 0$, a question posed by Erdős in 1981 [38]. In this section, we discuss known methods that have, or have the potential to, obtain asymptotic results for $R(C_4, K_m)$.

The Probabilistic Method

The main idea behind the probabilistic method (see e.g. [4]), which was originally pioneered by Erdős, involves proving that some mathematical structure exists by establishing that the *event* of such a structure existing occurs with positive probability. If there is a positive probability that something exists, we know that it does.

An important concept used in the probabilistic method is that of the *dependency digraph*. Given a collection of events $A_1, A_2, \ldots, A_n$ of some probability space Ω, the dependency directed graph D is defined as $V(D) = \{1, \ldots, n\}$ and $E(D) = \{(i, j) \mid A_i \text{ is not mutually independent of } A_j\}$. This graph is used in the well-known *Lovász Local Lemma*, a useful tool of the probabilistic method proven by Erdős and Lovász in 1975 [41]. The lemma is based on the following idea. If n mutually independent events hold with probability at least $p > 0$, then all events hold simultaneously with probability at least $p^n > 0$, an exponentially small number. A generalization of this fact is the situation when events are not mutually independent, but instead have "rare" dependencies. The goal is to still have certain events hold with a similarly small but positive probability. Indeed, this is possible, as presented in Lemma 1.

Lemma 1 (General Lovász Local Lemma, Erdős and Lovász, 1975 [41])**.** *Given events $A_1, A_2, \ldots, A_n$ of probability space Ω and the corresponding dependency digraph D, if there exists $x_1, \ldots, x_n$, $0 \le x_i < 1$, such that $\Pr(A_i) \le x_i \prod_{(i,j)\in E(D)}(1 - x_j)$ for all $i = 1, \ldots, n$, then*

$$\Pr\left(\bigwedge_{i=1}^{n} \overline{A_i}\right) \ge \prod_{i=1}^{n}(1 - x_i)$$

The core of the proof involves the use of induction to show that for any $S \subset \{1, \ldots, n\}$, $\Pr(A_i \mid \bigwedge_{j\in S} \overline{A_j}) \le x_i$, $i \notin S$.

This lemma was used by Spencer in 1977 to prove the lower bound of (3.1).

Theorem 8 (Spencer 1977, [134]). *There exists a positive constant c such that,*

$$R(C_4, K_n) \geq c\left(\frac{n}{\log n}\right)^{\frac{3}{2}}. \tag{3.2}$$

Proof. Consider an edge two-coloring of K_n where each edge is colored red and blue randomly and independently with probability p and $1-p$, respectively. For each set S of four vertices, let A_S be the event that S contains a red C_4. Likewise, for each set T of m vertices, let B_T be the event that all edges are blue. Clearly, $R(C_4, K_m) > n$ if and only if $\Pr(\bigwedge \overline{A_S} \wedge \bigwedge \overline{B_T}) > 0$.

Note that $\Pr(B_T) = (1-p)^{\binom{m}{2}}$ and $\Pr(A_S) \leq 6p^4$. Two events (either both from A_S, both from B_T, or one of each) are dependent if and only if the corresponding graphs share an edge. We can then construct dependency digraph D with vertices of all A_S and B_T, and connecting (in both directions) vertices according to this rule. Each A_S vertex is adjacent to $6\binom{n-4}{2} \leq n^2$ other $A_{S'}$ vertices and each B_T vertex is adjacent to $\binom{m}{2}\binom{n-m}{2} + \binom{m}{3}(n-m) \leq m^2n^2$ $A_{S'}$ vertices. Both A_S and B_T vertices are adjacent to at most $\binom{n}{m}$ $B_{T'}$ vertices.

The goal is now to find a suitable probability p, s, and t, all in the range $[0, 1)$, so that we meet the condition of Lemma 1. After plugging in the probabilities and dependencies of the events, the condition becomes

$$6p^4 \leq s(1-s)^{n^2}(1-t)^{\binom{n}{m}},$$

and

$$(1-p)^{\binom{m}{2}} \leq t(1-s)^{n^2}(1-t)^{\binom{n}{m}}.$$

After sophisticated analysis, Spencer determined that $p = c_1 n^{-2/3}$ and $m = c_2 n^{2/3} \log n$ work best, giving the lower bound (3.2). ∎

Spencer used a similar argument for bounding $R(C_3, K_m)$ (note $C_3 \cong K_3$), and in fact, his analysis for $R(C_4, K_m)$ is for graphs that avoid C_3 and C_4. It is unknown if avoiding C_3 matters for this analysis, or if a better probabilistic bound can be found that does not avoid C_3.

Recursive Constructions?

Another possible approach to constructing lower bounds of $R(C_4, K_m)$ is to do so without the probabilistic method. In 1993, Chung, Cleve, and Dagum [20] developed a recursive method in constructing lower bounds on $R(C_3, K_n)$. The main idea behind their approach is to take a small extremal graph, make copies of it, and join these copies in such a way that C_3 and a larger independent set are avoided. Specifically, let G be a C_3-free graph. The *fibration* of G is a graph H that contains six copies of G, say $G_0, \ldots, G_5$ with $v_i \in V(G_i)$ corresponding to $v \in V(G)$. They use six copies because edges $\{v_i, v_j\}$, $j = i + 1 \bmod 6$, and $\{v_i, v_k\}$, $k = i + 3 \bmod 6$, can clearly be in $E(H)$ without causing a triangle. Adding all such edges gives $\alpha(H) \leq 4\alpha(G)$.

They apply this idea of a fibration recursively. Take graphs $G_0, G_1, G_2, \ldots$ with $G_0 = C_5$ and G_{i+1} as the fibration of G_i. Then $|G_i| = 5 \cdot 6^i$ and $\alpha(G_i) \leq 2 \cdot 4^i$. Setting $i = \lfloor \log((n-1)/2)/\log 4 \rfloor$ yields $|G_i| > \frac{5}{6}((n-1)/2)^{\log 6/\log 4}$ and $\alpha(G_i) \leq n - 1$. Hence, $R(C_3, K_n) = \Omega(n^{\log 6/\log 4}) \approx \Omega(n^{1.29})$.

Interestingly, this bound is not as strong as the probabilistic bound of Spencer. No similar recursive construction exists for the lower bound of $R(C_4, K_n)$.

Related Parameters

The gap in the lower and upper bounds of (3.1) is intriguing in part because exact asymptotics are known for a number of related Ramsey numbers. In 1995, Kim [87] showed that for positive constant c, $R(C_3, K_m) \geq c(1 - o(1))m^2/\log m$, giving the lower bound in $R(C_3, K_m) = \Theta(m^2/\log m)$. Multicolored cycle-complete Ramsey numbers have also been studied. In 2005, Alon and Rödl [3] showed that $R(C_4, C_4, K_m) = \Theta(m^2 \text{poly} \log m)$ and $R(C_4, C_4, C_4, K_m) = \Theta(m^2/\log^2 m)$. They made use of a combination of spectral and probabilistic techniques applied to the Erdős-Rényi finite projective plane graphs, which we describe in Section 3.3.1. No similar techniques have been applied to $R(C_4, K_m)$ as successfully.

3.3 C_4-Free Graphs

It was beneficial for us to study certain properties of graphs that do not contain C_4 subgraphs, as they can often lead to bounds on the order of (C_4, K_m)-graphs. In particular, properties

that restrict the minimum degree and maximum size of C_4-free graphs proved useful in determining upper bounds of $R(C_4, K_m)$, as graphs breaking these restrictions contain a C_4. Generalized constructions of C_4-free graphs have provided lower bounds for related Ramsey problems, such as the cycle-star Ramsey numbers $R(C_4, K_{1,m})$.

Lemma 2, originally presented by Chvátal and Harary in 1972 [21], provides an upper bound for the minimum degree δ of a C_4-free graph.

Lemma 2 (Chvátal and Harary, 1972 [21])**.** *For all C_4-free graphs with order n and minimum degree δ,*

$$\delta^2 - \delta + 1 \leq n. \tag{3.3}$$

Proof. Let G be a C_4-free graph with minimum degree δ. Every pair of neighbors of $v \in V(G)$ are endpoints of a P_3, and therefore every vertex is the midpoint of at least $p = \binom{\delta}{2} = (\delta^2 - \delta)/2$ such paths. If there are p' total P_3 paths in G, it follows that $np \leq p'$. No two vertices of G can be the endpoints of two or more P_3's, so $p' \leq \binom{n}{2}$, and (3.3) follows. ■

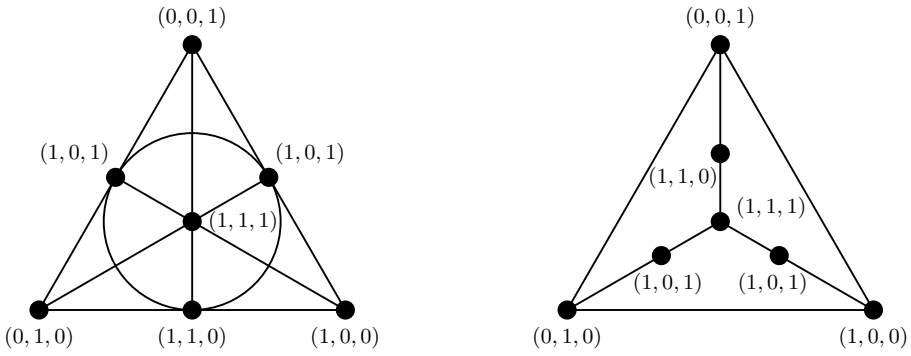
3.3.1 Finite Projective Planes

A well-known construction of C_4-free graphs, originally described by Erdős and Rényi in 1962 [42], makes use of polarities of finite projective planes. The C_4-free property of such graphs was explored simultaneously by Erdős, Rényi, and Sós [43] and Brown [13] in 1966.

Projective planes are planes in which all lines intersect at one point, and can be seen as a generalization of standard planes where all but parallel lines intersect (see e.g. [75]). A *finite projective plane* $PG(2, n)$ (see e.g. [95]) is a set of $n^2 + n + 1$ *points* and $n^2 + n + 1$ *lines* such that:

1. every line is incident to $n + 1$ points,
2. every point is incident to $n + 1$ lines,
3. every pair of lines intersect at exactly one point,
4. every pair of points share exactly one line.

This definition is often used to make the equivalence of lines and points apparent, as the two words can essentially be interchanged without altering the definition.



(a) $PG(2, q)$ (b) ER_2

Figure 3.2: The Fano plane $PG(2, q)$ and the C_4-free graph ER_2.

Finite projective planes are known to exist for all prime powers q and can be constructed from the finite field $GF(q)$. In such a construction, the points of $PG(2, q)$ are the triples (x, y, z), $x, y, z \in GF(q)$. For all $\alpha > 0 \in GF(q)$, (x, y, z) and $(\alpha x, \alpha y, \alpha z)$ are the same point, and the triple $(0, 0, 0)$ is excluded. Thus, there are $(q^3 - 1)/(q - 1) = q^2 + q + 1$ points. A line is constructed from all linear combinations of a pair of points. Note that this construction is equivalent to the set of all points $X = (x, y, z)$ which satisfy $ax + by + cz = 0$ for some a, b, and c in $GF(q)$. If we call this line $A = [a, b, c]$, there exists a mapping T from points to lines such that $TA = X$ and $TX = A$. This gives an "equivalence" between lines and points previously stated.

The construction of C_4-free graphs ER_q (commonly referred to as Erdős-Rényi graphs or ER-graphs) comes from mapping each point of $PG(2, q)$ to a vertex. Two vertices $v = (v_1, v_2, v_3)$ and $u = (u_1, u_2, u_3)$ are adjacent if and only if $v_1u_1 + v_2u_2 + v_3u_3 = 0$ in $GF(q)$. That is, $\{u, v\}$ is an edge if and only if point v is incident to the line Tu and, likewise, u is incident to Tv. Each vertex v of ER_q has degree q or $q + 1$, depending on whether $v_1^2 + v_2^2 + v_3^2 = 0$. Note that since every pair of points A and B are incident to exactly one line TC (item 4 from the definition), every pair of vertices will have only one common neighbor, and therefore ER_q is C_4-free.

Figure 3.2 shows the smallest non-trivial finite projective plane $PG(2, 2)$, with points and vertices labeled with the $GF(q)$ construction. This plane is known as the Fano plane, and is also a small combinatorial t-design as discussed in Section 4.3.2.

3.3.2 Turán Numbers for the Quadrilateral

Extremal problems involving C_4-free graphs provided insight into our study of $R(C_4, K_m)$. For a general graph H, let $\mathrm{ex}(n, H)$ denote the maximum number of edges an n-vertex, H-free graph can have; they are often called Turán numbers. In 1966, Erdős and Simonovits [44] showed that

$$\lim_{n\to\infty} \frac{\mathrm{ex}(n,H)}{n^2} = \frac{1}{2}\left(1 - \frac{1}{\chi(H)-1}\right), \tag{3.4}$$

which gives exact asymptotics for $\chi(H) \geq 3$. However, if H is bipartite, then (3.4) yields only $\mathrm{ex}(n, H) = o(n^2)$, making the asymptotics of this case distinct and open. For more information on related extremal problems in graph theory see [9]

The case of $H = C_{2k}$ seems to be particularly difficult. In 1982, it was conjectured by Erdős and Simonovitis [45] that asymptotically $\mathrm{ex}(n, C_{2k}) \sim \frac{1}{2}n^{1+1/k}$. Erdős, Rényi, and Sós [43] and Brown [13] used the ER-graphs to show this to be true for $H = C_4$, that is, that $\lim_{n\to\infty} \mathrm{ex}(n, C_4)/n^{3/2} = 1/2$. However, the conjecture was refuted for $k = 5$ by Lazebnik et. al in 1999 [98] and for $k = 3$ by Füredi et. al in 2006 [53].

Much less is known about the exact values of $\mathrm{ex}(n, C_4)$, as constructing witnesses for the lower bound seems to be difficult. It was conjectured by Erdős in 1973 [37] that $\mathrm{ex}(n, C_4) = \frac{1}{2}q(q+1)^2$ for $n = q^2 + q + 1$, q a prime power. That is, the graph ER_q has the maximum number of edges of a C_4-free graph with such an order. In 1996, Füredi [52] proved the conjecture for all $q > 13$. Unfortunately, no such formula exists for a general n. A well-known 1966 paper by Kövári, Sós, and Turán [89] states that for any $K_{2,m}$-free graph of order n and average degree d,

$$\binom{d}{2} \leq (m-1)\binom{n}{2}.$$

Applying this bound to $C_4 \cong K_{2,2}$ yields

$$\mathrm{ex}(n, C_4) \leq n\left(\frac{1+\sqrt{4n-3}}{4}\right). \tag{3.5}$$

This bound has proven useful in a number of Ramsey problems, including upper bounds for related Ramsey numbers by Caro et. al [16].

Values for $\mathrm{ex}(n, C_4)$ are known for all $n \leq 32$, and are presented in Table 3.2. The values

n	3	4	5	6	7	8	9	10	11	12
$\mathrm{ex}(n, C_4)$	3	4	6	7	9	11	13	16	18	21
n	13	14	15	16	17	18	19	20	21	22
$\mathrm{ex}(n, C_4)$	24	27	30	33	36	39	42	46	50	52
n	23	24	25	26	27	28	29	30	31	32
$\mathrm{ex}(n, C_4)$	56	59	63	67	71	76	80	85	90	92

Table 3.2: Known values for $\mathrm{ex}(n, C_4)$ [23, 142, 132]

for $n \leq 21$ were determined by Clapham, Flockhart, and Sheehan in 1989 [23] without the use of a computer. In 1992, Yuansheng and Rowlinson developed computational techniques to determine $\mathrm{ex}(n, C_4)$ and were able to, in addition to verifying Clapham et. al's work, compute values up to $n \leq 31$. In 2009, Shao, Xu, and Xu extended this method to compute $\mathrm{ex}(32, C_4) = 92$. The main idea behind this computational technique is based on the following observation. If a C_4-free graph G on n vertices is critical (that is, any additional edge causes a C_4), then $G - v$, $v \in V(G)$, may not be critical, but is a spanning subgraph of a critical $(n-1)$-vertex graph H. If $E \subset E(H)$ is picked such that $G - v = H - E$, then we can construct all critical G by attaching vertex v to $H - E$ in all possible ways. All such critical n-vertex graphs can be found using all possible H and Y.

3.4 Small Ramsey Numbers

Table 3.3 presents the known values and bounds for $R(C_4, K_m)$ for small m. In this section, we discuss some such results that were obtained without the use of a computer.

The witnesses for $R(C_4, K_3) > 6$ and $R(C_4, K_4) > 9$ can be obtained from a construction originally presented by Chvátal and Harary [21]. Consider a graph consisting of k disjoint triangles. Clearly, this graph does not contain a C_4 nor an independent set of order $k + 1$. Setting $k = 2, 3$ produces a (C_4, K_3)- and (C_4, K_4)-graph, respectively. This construction can be generalized to any connected graphs G_1 and G_2 with $|V(G_1)| = n$ and $\chi(G_2) = c$, where a (G_1, G_2)-graph is constructed as $c - 1$ disjoint copies of K_{n-1}.

We now summarize proofs of the upper bounds. A main concept common to both these proofs, as well as the computational approach we discuss in the next section, involves

m	$R(C_4, K_m)$	Year	References
3	7	1971	[18]
4	10	1972	[21]
5	14	1977	[22]
6	18	1987/1977	[47]/[129]
7	22	2002/1997	[123]/[78]
8	26	2002	[123]
9	30	this work	
10	36		
11	39–44		
12	42–53		

Table 3.3: Known values and bounds for $R(C_4, K_m)$.
Double references correspond to lower and upper bounds.

analyzing the neighborhoods of vertices of (C_4, K_m)-graphs. For a (C_4, K_m)-graph G with fixed $v \in V(G)$, let $X = N_G(v)$, $Y = V(G) - N_G[v]$, and X' and Y' be the graphs induced by X and Y, respectively. Clearly, X' does not contain the endpoints of a P_3 and $\alpha(Y') < m-1$. These simple observations prove to be useful in a number of situations, some of which we describe below.

$R(C_4, K_3) = R(C_3, C_4) = 7$ was given by Chartrand and Seymour in 1971 [18] as part of their proof that $R(C_3, C_m) = 2m - 1$ for all $m \geq 4$. To show $R(C_4, K_3) \leq 7$, consider a $(C_4, K_3; 7)$-graph G. From Lemma 2, $\delta(G) \leq 3$. Let $v \in V(G)$ have minimum degree and let $Y = V(G) \setminus N_G[v]$. If $\delta(G) = 2$, then $|Y| = 4$. To preserve $\alpha(G) < 3$, these 4 vertices must all be connected, but this causes a C_4. Therefore, $\delta(G) = 3$. The three vertices $N_G(v) = \{x_1, x_2, x_3\}$ must contain an edge, say $\{x_1, x_2\}$. $Y = \{y_1, y_2, y_3\}$ must be a triangle, or else v and the non-edge of Y cause $\alpha(G) \geq 3$. Since $\delta = 3$, x_1 and x_2 must both connect to a vertex in Y. If this vertex is the same, say y_1, then $\{v, x_1, y, x_2\}$ is a C_4. If they are distinct, say y_1 and y_2, then $\{x_1, y_1, y_2, x_2\}$ is a C_4. Thus, no $(C_4, K_3; 7)$-graph exists.

To show $R(C_4, K_4) \leq 10$, assume there exists a $(C_4, K_4; 10)$-graph G. From Lemma 2, $\delta(G) \leq 3$. If $\delta(G) \leq 2$ then $|Y| \geq 7$, which cannot be since $R(C_3, C_4) = 7$. Therefore, $\delta(G) = 3$ and $|Y| = 6$. Since Y cannot have an independent set of order 3, the graph induced by Y is either two disjoint triangles or two triangles connected with one edge. Let Y_1 and Y_2 be the former and latter of these graphs, respectively. No two vertices of

$X = \{x_1, x_2, x_3\} = N_G(v)$ can be adjacent to the same vertex in Y. If x_1, x_2, and x_3 are disjoint, then they must each connect to two and only two such vertices. Note that each edge of Y_1 and all but one edge of Y_2 is an end-point of a P_3. Therefore, if x_1 is adjacent to say y_1 and y_2, then $\{y_1, y_2\} \notin E(G)$ and $\{x_2, x_3, y_1, y_2\}$ is an independent set. A similar argument can be made for when $\{x_1, x_2\} \in E(G)$ and x_1 and x_2 are only connected to Y with one edge each.

The value $R(C_4, K_5) = 14$ was given by Clancy in 1977 [22] as part of a table that presented almost all $R(G, H)$ where $|V(G)| = 4$ and $|V(H)| = 5$. The value of $R(C_4, K_6)$ and bounds $21 \leq R(C_4, K_7) \leq 22$ were presented by Jayawardene and Rousseau in 1998 and 2000, respectively [78, 79]. The numbers $R(C_4, K_7)$, $R(C_4, K_8)$ and the bounds $30 \leq R(C_4, K_9) \leq 33$, $34 \leq R(C_4, K_{10}) \leq 40$ were given by Radziszowski and Tse in [123] in 2002. We discuss their methods in detail in the next section. Further upper bound improvements to 32 and 39 for $R(C_4, K_9)$ and $R(C_4, K_{10})$, respectively, were presented in [141].

3.5 Computational Approach

In 2002, Radziszowski and Tse [123] described a computational attack on $R(C_4, K_m)$ and determined the exact values for $m = 7$ and $m = 8$. The main goal behind the computations is to enumerate the sets $\mathcal{R}(C_4, K_m)$ completely. If $\mathcal{R}(C_4, K_m; n) \neq \emptyset$, then $R(C_4, K_m) > n$, and if $\mathcal{R}(C_4, K_m; n+1) = \emptyset$, then $R(C_4, K_m) \leq n + 1$. The latter is accomplished by extending $\mathcal{R}(C_4, K_m; t)$ to graphs in sets with higher m and/or t.

In this work, we were able to use these and similar computations to determine $R(C_4, K_9) = 30$ and $R(C_4, K_{10}) = 36$. Comparable algorithms have been used to find other Ramsey numbers, such as in [110, 57]. A description of these algorithms follows.

It is important to note that these algorithms were implemented independently by Ivan Livinsky [104] and most results that were obtained by our implementations were checked with his his. Our results always agreed.

3.5.1 Methods

Our enumeration of various classes of (C_4, K_m)-graphs uses two computational methods, VertexExtend and Glue, described below.

VertexExtend

This algorithm extends a $(C_4, K_m; n)$-graph G to all possible $(C_4, K_m; n+1)$-graphs G' containing G by attaching a new vertex v to all feasible neighborhoods in G. By feasible, we mean that the additional edges do not create a C_4 while also preserving $\alpha(G') < m$. If complexity of computations is ignored, then full enumeration of $\mathcal{R}(C_4, K_m; n+1)$ can clearly be obtained from $\mathcal{R}(C_4, K_m; n)$ with this method.

Glue

The second method, called the Glue algorithm, constructs $\mathcal{R}(C_4, K_m; n+\delta+1)$ from $\mathcal{R}(C_4, K_{m-1}; n)$, where δ is the minimum degree of the new graphs. For a $(C_4, K_m; n+\delta+1)$-graph G, let $v \in V(G)$ be such that $\deg_G(v) = \delta(G)$, and let X be the subgraph induced by $N_G(v)$; X must be a $(P_3, K_m; \delta)$-graph. Let Y be the induced subgraph of $V(G) \setminus (X \cup \{v\})$; Y must be a $(C_4, K_{m-1}; n)$-graph. If we know $\mathcal{R}(C_4, K_{m-1}; n)$, we can find all graphs in $\mathcal{R}(C_4, K_m; n+\delta+1)$ by considering how each vertex $x \in X$ can be connected to the vertices of Y. We call each neighborhood $N(x) \cap V(Y)$ the *cone* of x, denoted $c(x)$. We say that the cone $c(x)$ is *feasible* if:

1. $c(x)$ does not contain two endpoints of any P_3 in Y.
2. For distinct $x_1, x_2 \in V(X)$, $c(x_1) \cap c(x_2) = \emptyset$.
3. For each edge $\{x_1, x_2\} \in E(X)$, there is no $y_1 \in c(x_1)$ and $y_2 \in c(x_2)$ such that $\{y_1, y_2\} \in E(Y)$.
4. For each subgraph induced by $X' \subseteq X$ and Y' induced by $V(Y) \setminus \bigcup_{x \in X'} c(x)$, $\alpha(X') + \alpha(Y') < m$.

Conditions 1, 2, and 3 prevent C_4's, while condition 4 prevents independent sets of order m. Figure 3.3 displays the main idea behind Glue, while Figure 3.4 shows how a $(C_4, K_5; 13)$-graph can be glued from the $(C_4, K_4; 9)$-graph of Figure 3.1.

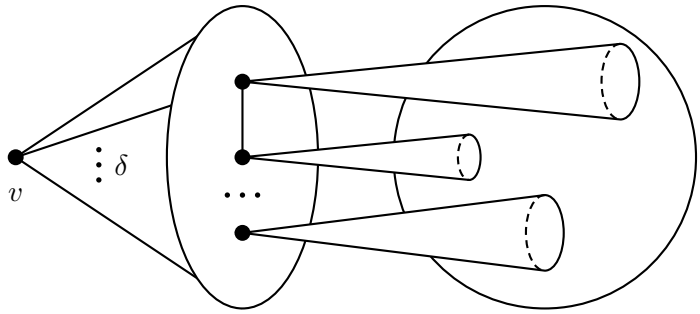


Figure 3.3: Gluing to a $(C_4, K_m; \delta + t + 1)$-graph.

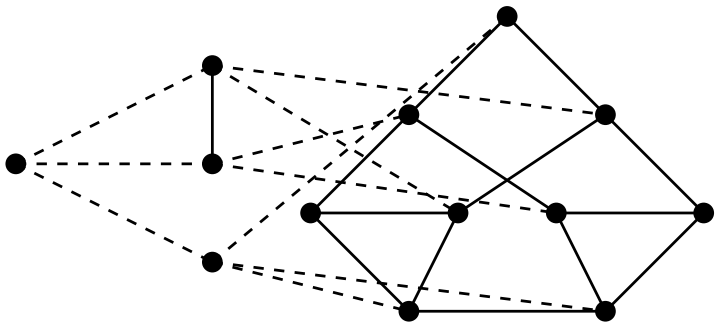


Figure 3.4: Gluing a $(C_4, K_4; 9)$-graph to a $(C_4, K_5; 13)$-graph.

3.5.2 Implementation and Optimization

As previously mentioned, our implementations of VertexExtend and Glue were used in conjunction with those of Livinsky in order to corroborate the correctness of the results. In all cases where both implementations were used, the results agreed.

The rules for gluing (C_4, K_m)-graphs described in Section 3.5.1 allowed for a much needed speedup in computations. In most cases, it was beneficial to preprocess the Y graphs before gluing, storing information about the feasibility of the cones. For example, all subsets of vertices containing endpoints of a P_3 were removed from the list of feasible cones. Speed was greatly increased by precomputing the independence number $\alpha(Y')$ of each subgraph, which was critical for efficient testing of condition 4. This proved to be a bottleneck of the computations, and multiple strategies and implementations were tested. The most efficient algorithm implemented was based on *Algorithm 1: Precomputing independence number*, described in [57]. All data was stored in arrays of size 2^n, where the integer index of the array represented the bit-set of the vertices of the subgraph.

Algorithm 3.5.1: Glue(δ, $\Gamma \subseteq \mathcal{R}(C_4, K_{m-1})$)

---

for each $G \in \Gamma$

do
- $n \leftarrow |V(G)|$
- Find all $S \in V(G)$ without P_3 endpoints (2^n `Boolean` array)
- Compute $\alpha(T)$ for all $T \subseteq G$ (2^n `int` array)
- Find each neighborhood of $T \subseteq G$ (2^n `int` array)
- **for each** P_3-free graph X, $|V(X)| = \delta$
  - **do**
    - Recursively find all valid $c(x)$, $x \in V(X)$
    - Store glued graph in L
- Remove isomorphisms in L

return (L)

In order to test isomorphisms, we made use of the well-known software `nauty` by Brendan McKay [109]. Livinsky implemented an algorithm described by William Kocay [88].

3.6 New Results

Our first result was a full enumeration of $\mathcal{R}(C_4, K_7)$. This was significant, as the same enumeration was computationally infeasible when these methods were attempted in 2002

[123]. $\mathcal{R}(C_4, K_7)$ was first obtained using VertexExtend. The same results were obtained when gluing from $\mathcal{R}(C_4, K_6)$. The statistics of $\mathcal{R}(C_4, K_7)$ by vertex and edge counts are displayed in Tables 3.4 and 3.5. The cases of counts found in [123] agree with ours.

Once $\mathcal{R}(C_4, K_7)$ was obtained, we were able to construct $\mathcal{R}(C_4, K_8; n)$ for n equal to 23, 24, and 25. The gluing of $\mathcal{R}(C_4, K_8; 23)$ turned out to be the most computationally expensive, as there are 353015495 such graphs, but was needed in order to extend them further to $\mathcal{R}(C_4, K_9; 29)$. The counts for $\mathcal{R}(C_4, K_8; 23)$ are displayed by size and minimum degree in Table 3.6. Statistics for $\mathcal{R}(C_4, K_8; 24)$ and $\mathcal{R}(C_4, K_8; 25)$ are gathered in Table 3.7. Our computations found that no (C_4, K_8)-graph exists with minimum degree 5.

3.6.1 $R(C_4, K_9)$

We constructed the sets $\mathcal{R}(C_4, K_9; 29)$ and $\mathcal{R}(C_4, K_9; 30)$ with the Glue algorithm. Since $R(C_4, K_8) = 26$, any $(C_4, K_9; 29)$-graph has minimum degree 3, 4, or 5 and can be obtained from $\mathcal{R}(C_4, K_8; n)$ for $n = 25, 24, 23$ by Glue. Note that the minimum degree of a $(C_4, K_8; 23)$-graph must be 4 in order to glue to a graph of minimum degree 5. This restriction improved the speed of computation, as there are a large number of $(C_4, K_8; 23)$-graphs to consider. Statistics for $\mathcal{R}(C_4, K_9; 29)$ are found in Table 3.8.

Similarly, any $(C_4, K_9; 30)$-graph has minimum degree 4 or 5, and can be obtained from $\mathcal{R}(C_4, K_8; 25)$ or $\mathcal{R}(C_4, K_8; 24)$, respectively, via Glue. No $(C_4, K_9; 30)$-graphs were found, resulting in the following theorem.

Theorem 9. $R(C_4, K_9) = 30$.

3.6.2 $R(C_4, K_{10})$

Theorem 10. $R(C_4, K_{10}) = 36$.

Proof. Livinsky [104] found two 6-regular $(C_4, K_{10}; 35)$-graphs H_1 and H_2, establishing the lower bound. The orbits of H_1 are depicted in Figure 3.5 and its adjacency matrix is presented in Figure 3.6.

In order to prove $R(C_4, K_{10}) \leq 36$, it is necessary to show that no $(C_4, K_{10}; 36)$-graph exists. As $R(C_4, K_9) = 30$, from Lemma 2, a $(C_4, K_{10}; 36)$-graph has minimum degree at

n / e	7	8	9	10	11	12	13	14	15
1	1								
2	2	1							
3	5	4	1						
4	9	9	4	1					
5	18	20	14	4	1				
6	29	42	40	16	3	1			
7	30	71	91	57	13	2			
8	17	88	178	172	56	9	1		
9	5	72	274	422	221	41	4		
10		31	289	805	737	183	19	1	
11		5	197	1135	1947	779	94	5	
12			74	1097	3861	2912	469	28	1
13			10	670	5405	8660	2221	151	5
14				222	5046	18943	9455	826	29
15				34	2965	28496	32805	4367	163
16				2	971	27902	84467	21211	920
17					146	16897	148686	87187	5218
18					11	5831	168441	277608	27740
19						1013	116266	622072	130043
20						82	45788	904916	507036
21						3	9434	801944	1513611
22							916	406222	3119854
23							39	108749	4033237
24							2	14039	3021620
25								818	1215627
26								24	241075
27								1	21639
28									851
29									22
30									2
Total	116	343	1172	4637	21383	111754	619107	3250169	13838693

Table 3.4: Statistics for $\mathcal{R}(C_4, K_7; n)$, $7 \leq n \leq 15$.
Note that for $n < 7$ the counts would be for all C_4-free graphs.

n / e	16	17	18	19	20	21
14	1					
15	5					
16	23	1				
17	116	3				
18	644	11	1			
19	3602	51	1			
20	19588	251	3			
21	97521	1311	12			
22	423964	6805	45			
23	1543985	33476	198			
24	4434855	149441	908			
25	9068568	585687	4045			
26	11612126	1964782	16971			
27	8299450	5448131	64462			
28	3016205	11583843	219831			
29	511367	16465694	672324	1		
30	37318	13277929	1813931	18		
31	1167	5287770	4096321	233		
32	26	938464	6953952	2399		
33	2	68369	7533349	17474		
34		2018	4275886	83786		
35		35	1064229	261093		
36		1	102512	520551		
37			3512	605219	1	
38			53	328849	12	
39			1	64919	126	
40				4132	999	
41				107	3611	
42				4	3762	
43					897	
44					53	
45					2	1
46						2
Total	39070533	55814073	26822547	1888785	9463	3

Table 3.5: Statistics for $\mathcal{R}(C_4, K_7; n)$, $16 \leq n \leq 21$.

δ / e	1	2	3	4	Total
40		1			1
41		13			13
42		201			201
43		3055	108		3163
44		36884	8517		45401
45		302179	260678		562857
46	1	1449548	3502385	83	4952017
47	6	3662039	23059729	35368	26757142
48	29	4576213	75076644	1563123	81216009
49	53	2716695	110589375	11348103	124654226
50	27	744258	66302337	19535975	86582597
51	3	95358	15327155	9727032	25149548
52		5827	1352590	1588719	2947136
53		164	47152	94684	142000
54		6	732	2404	3142
55			4	37	41
56				1	1
Total	119	13592441	295527406	43895529	353015495

Table 3.6: Size vs minimum degree of graphs in $\mathcal{R}(C_4, K_8; 23)$.
All such graphs with $\delta = 4$ were used with GLUE to find $(C_4, K_9; 29)$-graphs.

e \ n	24	25
48	1	
49	6	
50	48	
51	394	
52	3133	
53	21116	
54	60646	
55	57944	
56	18863	
57	2102	
58	96	2
59	4	10
60		15
61		9
Total	164353	36

Table 3.7: Statistics for $\mathcal{R}(C_4, K_8; n)$, $n = 24, 25$.
All such graphs were used with GLUE to find $(C_4, K_9; 29)$-graphs and to show that no $(C_4, K_9; 30)$-graphs exist.

e \ δ	3	4	5	Total
70	1	1		2
71	8	5		13
72	12	11		23
73	18	33	1	52
74	10	64	7	81
75		49	9	58
76		19	7	26
77		6	4	10
78			2	2
Total	49	188	30	267

Table 3.8: Size vs minimum degree of graphs in $\mathcal{R}(C_4, K_9; 29)$.
All such graphs were used during GLUE computations to show that no $(C_4, K_{10}; 36)$-graph exists.

most 6 and can be obtained from gluing a $(C_4, K_9; 29)$-graph. Gluing all of $\mathcal{R}(C_4, K_9; 29)$ resulted in finding no such graphs. ■

The automorphism group $\mathrm{Aut}(H_1)$ has order 24 and its action on $V(H_1)$ has four orbits of 24, 6, 4, and 1 vertices, respectively. The automorphism group $\mathrm{Aut}(H_2)$ has order 40 and its action on $V(H_2)$ has four orbits of 20, 10, and 5 vertices. Both graphs H_1 and H_2 have 105 edges and 35 triangles, with each vertex on three triangles. They are both bicritical: removing any edge produces an independent set of order 10, and adding any edge produces a C_4.

Interestingly, no $(C_4, K_{10}; n)$-graphs for $n = 34, 35$ were obtained by gluing from $\mathcal{R}(C_4, K_9; 29)$.

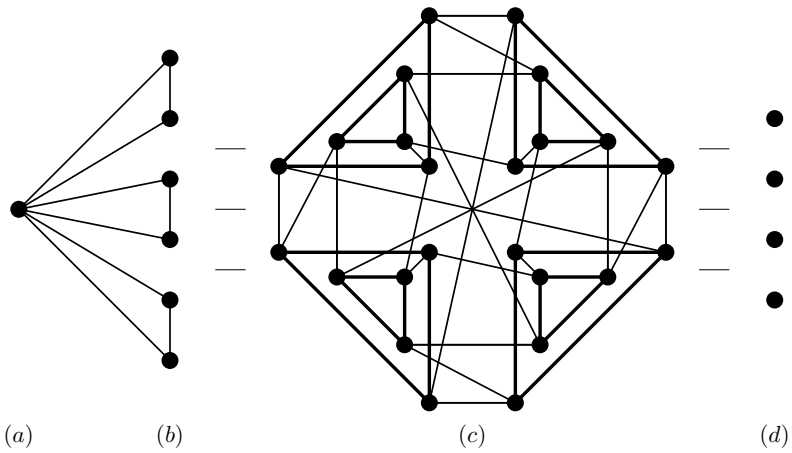


Figure 3.5: The four orbits of $\mathrm{Aut}(H_1)$. Parts (b) and (c) are connected by 24 edges, as well as (c) and (d).

3.6.3 Higher Parameters

Theorem 11. $39 \leq R(C_4, K_{11}) \leq 44$.

Proof. The lower bound is obtained by construction. A $(C_4, K_{11}; 38)$-graph can easily be obtained by adding a triangle to H_1 or H_2.

If a $(C_4, K_{11}; 44)$-graph G exists, then from Lemma 2 it follows that G must have minimum degree at most 7. Such a graph can be obtained by applying Glue to a $(C_4, K_{10}; 36)$-graph. However, since $R(C_4, K_{10}) = 36$, no such graph exists, and therefore G does not exist as

	a	b						c																								d			
a	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
b	1	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0
	1	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0
	1	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0
	1	0	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0
	1	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0
c	0	1	0	0	0	0	0	0	1	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	1	0	0	0
	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1
	0	0	1	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0
	0	0	0	1	0	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0
	0	0	0	0	0	0	1	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0
	0	1	0	0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0
	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
	0	0	0	0	0	1	0	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0
	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0
	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0
	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	1	0	0	0	0	0	1	0	0	0
	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0	1	0	0
	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	1	0
	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	1	0
	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	1
	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	0	0	0
	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	1	0	0	0	0	0	0	1
	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0
	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1	0	0
	0	0	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1
	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0
	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	1	0	0	1	0	0
d	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	1	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0	1	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	0	1	0	1	0	0	1	0	0	0	0	0	0

Figure 3.6: Adjacency matrix of H_1 separated by the orbits of $\mathrm{Aut}(H_1)$.

well. ■

Theorem 12. $42 \leq R(C_4, K_{12}) \leq 52$.

Proof. The lower bound is obtained similarly as before, by adding a triangle to the $(C_4, K_{11}; 38)$-graphs of Theorem 11.

As $R(C_4, K_{11}) \leq 44$, any (C_4, K_{12})-graph can be obtained by applying Glue to a (C_4, K_{11})-graph with order at most 43. From Lemma 2, such a graph must have a minimum degree of at most 7, and therefore an order of at most 51. Thus, $R(C_4, K_{12}) \leq 52$. ■

Chapter 4

LLL Algorithm

4.1 Introduction to Lattices

Let $\mathcal{L}$ be a subset of vectors of $\mathbb{R}^n$. $\mathcal{L}$ is a *lattice* if there exists a collection of vectors $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n \in \mathbb{R}^n$ such that $\mathcal{L}$ is exactly all linear combinations of $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n$ with integer coefficients. That is, $[\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n]$ is the basis of $\mathcal{L}$, or

$$\mathcal{L} = \text{span}_{\mathbb{Z}}(\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n) = \left\{ \sum_{i=1}^{n} x_i \mathbf{b}_i \mid x_i \in \mathbb{Z}, \text{ for } 1 \leq i \leq n \right\}. \tag{4.1}$$

In matrix form, (4.1) is simply $\mathcal{L}(B) = \{B\mathbf{x} \mid \mathbf{x} \in \mathbb{Z}^n\}$. Lattices are graphically represented as an infinite grid of connected points in n dimensions. Two examples of lattices in $\mathbb{R}^2$ are presented in Figure 4.1. In group theory, lattices are discrete additive subgroups of $\mathbb{R}^n$, with $\text{span}(\mathcal{L}) = \mathbb{R}^n$ for any lattice $\mathcal{L}$.

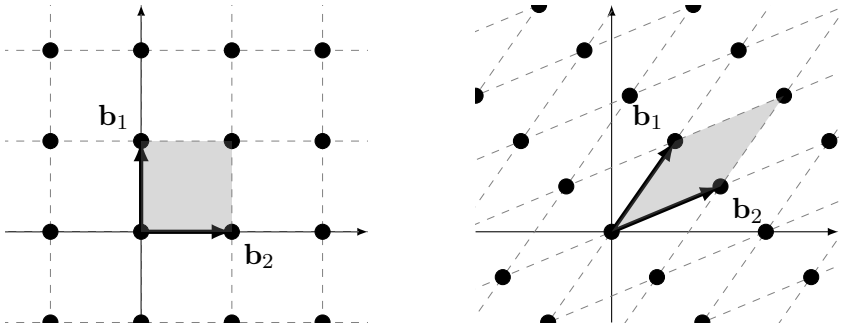


Figure 4.1: Examples of lattices in $\mathbb{R}^2$

Lattices play significant roles in a number of areas of mathematics, including geometry, number theory, group theory, and coding theory. They have gained particular attention in the past 20 years for their applications to cryptography, as certain properties of them are

computationally difficult to determine (see for example [111] and Chapter 14 of [115]). The study of lattice-based cryptosystems was set in motion by Ajtai's breakthrough result of 1996 [1], which found a connection between the worst-case and average-case hardness of one such problem that is now described.

The SHORTEST-VECTOR problem (SVP) asks to find the shortest non-zero vector of a lattice. That is, given a basis B of lattice $\mathcal{L}$, determine $B\mathbf{x}$ such that $\|B\mathbf{x}\| \leq \|B\mathbf{y}\|$ for any non-zero $\mathbf{y} \in \mathbb{Z}^n$. The length of such a vector is commonly denoted $\lambda(\mathcal{L})$. The CLOSEST-VECTOR problem (CVP) is similar, asking for the closest vector to a given vector $\mathbf{t}$ of a lattice. Decision versions of both problems are known to be **NP**-complete, and it can be shown that SVP is not harder than CVP (see [111]). Polynomial time $f(n)$-approximation algorithms are known for exponential $f(n)$; such approximation with $f(n) = n^c$ for some constant c remain open; and in 2005, Khot [86] showed that SVP is **NP**-hard to approximate when $f(n)$ is constant.

Classical texts on lattices include those of Cassels from 1971 [17] and Gruber and Lekkerkerker from 1987 [64]. Most of the background presented in this and the next section is found in [111, 84, 93, 115].

A major breakthrough in the computational study of lattices occured in 1982 when Lenstra, Lenstra, and Lovász [99] discovered what is now known as the LLL Algorithm. The algorithm, when given a basis of lattice, returns a relatively short basis, called a *reduced basis*, in polynomial time. Although its theoretical worst case results are poor, the algorithm tends to be a remarkable success in practice. It has been applied to a number of areas and related problems, including integer programming, factoring polynomials, approximating SVP and CVP, cryptanalysis, and **NP**-hard search problems.

The focus of this chapter is on these applications, and in particular, ways the LLL algorithm can be used to attack hard combinatorial search problems. What follows is the concept of an orthogonal basis, a key part of the LLL Algorithm, and the useful properties of lattices we can determine from them.

4.1.1 Gram-Schmidt and Orthogonal Bases

Vectors $\mathbf{x}$ and $\mathbf{y}$ are *orthogonal* if $\mathbf{x} \cdot \mathbf{y} = 0$, and a basis $B \in \mathbb{R}^n$ is an *orthogonal basis* if all vectors of B are pairwise orthogonal. Given basis B of subspace $S \in \mathbb{R}^n$, the Gram-Schmidt

process (see [138, 93]) returns an orthogonal basis B^* of S. The algorithm is based on the fact that $\mathbf{x}$ and $\mathbf{y}^* = \mathbf{y} - \text{proj}_{\mathbf{x}}(\mathbf{y})$ are orthogonal, where $\text{proj}_{\mathbf{x}}(\mathbf{y})$ is the *projection* of $\mathbf{y}$ onto $\mathbf{x}$, defined as

$$\text{proj}_{\mathbf{x}}(\mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\|^2}\mathbf{x}.$$

Given $B = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n]$, the Gram-Schmidt process returns the orthogonal basis $B^* = [\mathbf{b}_1^*, \mathbf{b}_2^*, \ldots, \mathbf{b}_n^*]$ as follows:

$$\begin{aligned}
\mathbf{b}_1^* &= \mathbf{b}_1, \\
\mathbf{b}_2^* &= \mathbf{b}_2 - \text{proj}_{\mathbf{b}_1^*}(\mathbf{b}_2), \\
\mathbf{b}_3^* &= \mathbf{b}_3 - \text{proj}_{\mathbf{b}_1^*}(\mathbf{b}_3) - \text{proj}_{\mathbf{b}_2^*}(\mathbf{b}_3), \\
&\vdots \\
\mathbf{b}_n^* &= \mathbf{b}_n - \sum_{i=1}^{n-1} \text{proj}_{\mathbf{b}_i^*}(\mathbf{b}_n).
\end{aligned}$$

The process is presented in Algorithm 4.1.1. Given k vectors of length n, the algorithm returns k orthogonal vectors in $O(nk^2)$ time. When the basis is full-rank, the running time is $O(n^3)$.

Algorithm 4.1.1: GRAM-SCHMIDT($B = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n]$)

$\mathbf{b}_1^* \leftarrow \mathbf{b}_1$

for $j \leftarrow 2$ **to** n

do
- $\mathbf{b}_j^* \leftarrow \mathbf{b}_j$
- **for** $i \leftarrow 1$ **to** $j - 1$
  - **do**
    - $\mu_{j,i} \leftarrow (\mathbf{b}_i^* \cdot \mathbf{b}_j)/\|\mathbf{b}_i^*\|^2$
    - $\mathbf{b}_j^* \leftarrow \mathbf{b}_j^* - \mu_{i,j}\mathbf{b}_i^*$

return $([\mathbf{b}_1^*, \mathbf{b}_2^*, \ldots, \mathbf{b}_n^*], \{\mu_{i,j} \mid 1 \leq j < i \leq n\})$

It is important to note that lattices, unlike subspaces of $\mathbb{R}^n$, do not necessarily have orthogonal bases. Given basis B of lattice $\mathcal{L}$, we can still apply Gram-Schmidt and obtain

B^*, but this will most likely not be a basis of $\mathcal{L}$.

The *fundamental parallelepiped* of basis B is defined as $P(B) = \{B\mathbf{x} \mid \mathbf{x} \in [0,1)^n\}$, and is shown as the shaded regions in Figure 4.1. The volume of this parallelepiped is known as the *volume* of lattice $\mathcal{L}(B)$, denoted $\mathrm{vol}(\mathcal{L})$. The parallelepipeds of any two bases of $\mathcal{L}$ have equal volumes, defined as

$$\mathrm{vol}(\mathcal{L}) = \prod_{i=1}^{n} \|\mathbf{b}_i^*\|.$$

When $\mathcal{L}$ is full-rank, we obtain $\mathrm{vol}(\mathcal{L}) = |\det B| = |\det B^*|$.

This equality is apparent when considering the Gram-Schmidt process. From Algorithm 4.1.1, setting $\mu_{ij} = (\mathbf{b}_i \cdot \mathbf{b}_j^*)/\|\mathbf{b}_j^*\|^2$ yields $\mu_{ii} = 1$ and

$$\mathbf{b}_i = \sum_{j=1}^{i} \mu_{ij}\mathbf{b}_j^*. \tag{4.2}$$

Define matrix A with entries a_{ij} as

$$a_{ij} = \begin{cases} \mu_{ij} & \text{if } i \geq j, \\ 0 & \text{otherwise,} \end{cases}$$

then A is a lower triangular matrix with 1's on the diagonal. Equation (4.2) becomes the matrix equation $B = AB^*$, and since $\det A = 1$, we obtain $\det B = \det AB^* = \det B^*$.

It will be useful for us to define a similar parameter to $\mathrm{vol}(\mathcal{L})$ for a basis that may not be orthogonal. The *weight* of a basis B is defined as:

$$\mathrm{wt}(B) = \prod_{j=1}^{n} \|\mathbf{b}_i\|.$$

It is known through Hadamard's inequality that $\mathrm{vol}(\mathcal{L}) \leq \mathrm{wt}(B)$, with equality achieved only when B is an orthogonal basis or when one of the columns is $\mathbf{0}$. As reducing B is seen as making the basis "nearly orthogonal," it is our goal to make the ratio $\mathrm{wt}(B)/\mathrm{vol}(\mathcal{L})$ as close to 1 as possible.

Another result obtained from orthogonal bases is presented in Lemma 3, which gives a lower bound on SVP.

Lemma 3 (Lenstra, Lenstra, Lovász, 1982 [99]). *Given lattice $\mathcal{L}$ with basis $B = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n]$, and $B^* = [\mathbf{b}_1^*, \mathbf{b}_2^*, \ldots, \mathbf{b}_n^*]$ obtained from Algorithm 4.1.1,*

$$\|\mathbf{b}\| \geq \min_j \|\mathbf{b}_j^*\|,$$

for any nonzero $\mathbf{b} \in \mathcal{L}$.

Proof. Since $\mathbf{b} \in \mathcal{L}$, it follows that there exists some $k \leq n$ such that $\mathbf{b} = \sum_{j=1}^{k} \mathbf{b}_j z_j$ where $z_j \in \mathbb{Z}$. From (4.2) we obtain $\mathbf{b} = \sum_{i=1}^{k} z_i \sum_{j=1}^{i} \mu_{ij} \mathbf{b}_i^*$ and since $\mu_{kk} = 1$, this becomes $\mathbf{b} = z_k \mathbf{b}_k^* + \sum_{i=1}^{k-1} \sum_{j=1}^{i} z_i \mu_{ij} \mathbf{b}_i^*$ or

$$\mathbf{b} = z_k \mathbf{b}_k^* + \sum_{i=1}^{k-1} z_i^* \mathbf{b}_i^*,$$

with $z_i^* \in \mathbb{R}$. Then, because of orthogonality,

$$\begin{aligned} \|\mathbf{b}\| &= |z_k| \, \|\mathbf{b}_k^*\| + \left(\sum_{i=1}^{k-1} (z_i^*)^2 (\mathbf{b}_i \cdot \mathbf{b}_i^*) \right)^{\frac{1}{2}} \\ &\geq |z_k| \, \|\mathbf{b}_k^*\| \\ &\geq \min_j \|\mathbf{b}_j^*\| \end{aligned}$$

■

4.2 Reducing the Basis

The goal of the LLL Algorithm is to find a basis of a lattice that is "nearly orthogonal." This is accomplished through the definition of a reduced basis. Let B be an ordered basis of lattice $\mathcal{L}$ and $(B^*, (\mu_{ij})) = \text{GRAM-SCHMIDT}(B)$. Then, B is a *reduced* (or y-reduced) basis of $\mathcal{L}$ if:

1. $|\mu_{i,j}| \leq \frac{1}{2}$ for all $i < j$, and
2. $\|\mathbf{b}_{j+1}^* + \mu_{j+1,j} \mathbf{b}_j^*\|^2 \geq y \|\mathbf{b}_j^*\|^2$ for $1 \leq j \leq n-1$ and $\frac{1}{4} < y < 1$.

Let λ_1 be the shortest vector length of lattice $\mathcal{L}$. Setting $y = 3/4$ results in the following theorem:

Theorem 13 (Lenstra, Lenstra, Lovász, 1982 [99]). *Let $B = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n]$ be a reduced basis of $\mathcal{L}$. Then,*

$$\|\mathbf{b}_1\| \leq 2^{\frac{n-1}{2}} \lambda_1. \tag{4.3}$$

Proof. From the definition of a reduced basis, we have

$$\begin{aligned} \frac{3}{4}\|\mathbf{b}_j^*\|^2 &\leq \|\mathbf{b}_{j+1}^* + \mu_{j+1,j}\mathbf{b}_j^*\|^2 \\ &= \|\mathbf{b}_{j+1}^*\|^2 + 2\mu_{j+1,j}(\mathbf{b}_{j+1}^* \cdot \mathbf{b}_j^*) + \mu_{j+1,j}^2\|\mathbf{b}_j^*\|^2 \\ &= \|\mathbf{b}_{j+1}^*\|^2 + \mu_{j+1,j}^2\|\mathbf{b}_j^*\|^2 \\ &\leq \|\mathbf{b}_{j+1}^*\|^2 + \frac{1}{4}\|\mathbf{b}_j^*\|^2 \\ \implies \quad \frac{1}{2}\|\mathbf{b}_j^*\|^2 &\leq \|\mathbf{b}_{j+1}^*\|^2. \end{aligned} \tag{4.4}$$

Since $\mathbf{b}_1^* = \mathbf{b}_1$, iterating the inequality $\|\mathbf{b}_j^*\|^2 \leq 2\|\mathbf{b}_{j+1}^*\|$ gives $\|\mathbf{b}_1\|^2 \leq 2^{j-1}\|\mathbf{b}_j^*\|^2$. From Lemma 3 it follows that $\lambda_1 \geq \min_j \|\mathbf{b}_j^*\|$. Then,

$$\begin{aligned} \|\mathbf{b}_1\|^2 &\leq \min_j \{2^{j-1}\|\mathbf{b}_j^*\|^2\} \\ &\leq 2^{n-1} \min_j \|\mathbf{b}_j^*\|^2 \\ &\leq 2^{n-1}{\lambda_1}^2, \end{aligned}$$

and we obtain (4.3). ∎

A combination of $\|\mathbf{b}_i^*\|^2 \leq 2^{j-i}\|\mathbf{b}_j^*\|^2$ from above, and $\mathbf{b}_j = \sum_{i=1}^{j} \mu_{ji}\mathbf{b}_i^*$ from (4.2) gives:

$$\|\mathbf{b}_j\|^2 = \|\mathbf{b}_j^*\|^2 + \sum_{i=1}^{j-1} \mu_{ji}^2\|\mathbf{b}_i^*\|^2 \leq \|\mathbf{b}_j^*\|^2 + \|\mathbf{b}_j^*\|^2 \sum_{i=1}^{j-1} 2^{j-i-2} \leq 2^{j-1}\|\mathbf{b}_j^*\|^2,$$

which results in the following Corollary.

Corollary 13.1 (Lenstra, Lenstra, Lovász, 1982 [99]). *Let $B = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n]$ be a reduced basis of $\mathcal{L}$. Then,*

$$\mathrm{wt}(B) \leq 2^{\frac{n(n-1)}{4}} \mathrm{vol}(\mathcal{L}) \tag{4.5}$$

Note that line (4.4) is another definition of a reduced basis: When $y = 3/4$, for any two consecutive vectors $\mathbf{b}_j$ and $\mathbf{b}_{j+1}$, the length of $\mathbf{b}_{j+1}$ must be at most $\sqrt{2}$ times larger than that of $\mathbf{b}_j$.

4.2.1 The Algorithm

The Lenstra-Lenstra-Lovász (LLL) algorithm [99] is a well-known algorithm that produces a reduced basis of a lattice in polynomial time. The algorithm, named after its three authors, was invented in 1982 to be used as a subroutine for a polynomial-time algorithm that factored polynomials in $\mathbb{Q}[x]$. It has since been recognized as a significant achievement in computer science, and is considered part of the foundation of the computational study of lattices. Perhaps even more remarkable than its numerous applications is its simplistic and adaptable design.

The main idea behind the algorithm is intuitive: Run Gram-Schmidt and step through the basis, checking each part against the rules of the definition. If some part fails a rule, fix that part, and start over. Then, repeat until a reduced basis is found. The algorithm uses two main linear transformations to achieve this:

1. If $|\mu_{i,j}| > \frac{1}{2}$, then set $\mathbf{b}_j = \mathbf{b}_j - \left\lfloor \mu_{i,j} + \frac{1}{2} \right\rfloor \mathbf{b}_i$ for some $1 \leq i < j \leq n$.
2. If $\|\mathbf{b}^*_{j+1} + \mu_{j,j+1}\mathbf{b}^*_j\|^2 < \frac{3}{4}\|\mathbf{b}^*_j\|^2$ for some $1 \leq j < n$, swap $\mathbf{b}_j$ and $\mathbf{b}_{j+1}$.

According to Lenstra [100], it does not matter how these two transformations are arranged, as after a finite number of steps, they will produce a reduced basis. Algorithm 4.2.1 presents one such arrangement. It can be shown that this algorithm terminates in polynomial time. In fact, the running time is $O(n^5 \log(\max_{1 \leq i \leq n} \|\mathbf{b}_i\|^2))$ (see e.g. [93, 84]).

4.2.2 Weight Reduction

Sometimes applying LLL to a basis is not enough to obtain the desired results, as the reduced basis still does not contain short enough vectors. A simple strategy to overcome this involves decreasing the weight of a basis $\mathrm{wt}(B)$.

We discuss a method for decreasing the weight of a basis developed by Kreher and Radziszowski, as described in [93]. This method was used by them in [92] to find simple t-designs and in [122] to attack instances of SUBSET-SUM. The main idea is simple: pick two vectors in the basis, and if their sum (or difference) is shorter than the longer of the two, replace it. That is, we search basis B for vectors $\mathbf{b}_i$ and $\mathbf{b}_j$ such that

$$\|\mathbf{b}_i + \epsilon \mathbf{b}_j\| < \max\{\|\mathbf{b}_i\|, \|\mathbf{b}_j\|\}$$

Algorithm 4.2.1: LLL($B = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n]$)

$([\mathbf{b}_1^*, \mathbf{b}_2^*, \ldots, \mathbf{b}_n^*], \{\mu_{j,i}|_{1 \le i < j \le n}\}) \leftarrow$ GRAM-SCHMIDT$(\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n)$

Step 1

for $j \leftarrow 2$ **to** n

do { **for** $i \leftarrow j-1$ **downto** 1

do { **if** $|\mu_{ij}| > 1/2$

then $\mathbf{b}_j \leftarrow \mathbf{b}_j - \lfloor \mu_{i,j} + \frac{1}{2} \rfloor \mathbf{b}_i$

Step 2

$([\mathbf{b}_1^*, \mathbf{b}_2^*, \ldots, \mathbf{b}_n^*], \{\mu_{i,j}|_{1 \le i < j \le n}\}) \leftarrow$ GRAM-SCHMIDT$(\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n)$

if there exists j s.t. $\|\mathbf{b}_{j+1}^* + \mu_{j+1,j}\mathbf{b}_j^*\|^2 < \frac{3}{4}\|\mathbf{b}_j^*\|^2$

then { swap $\mathbf{b}_j$ and $\mathbf{b}_{j+1}$

go to Step 1

return $([\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n])$

for $\epsilon \in \{-1, 1\}$.

This can be accomplished in $O(n^3)$ time when making the following observation. Let $\mathbf{b}_i$ and $\mathbf{b}_j$ be vectors in our reduced basis such that $\mathbf{v} = \mathbf{b}_i + e\mathbf{b}_j$, $\epsilon \in \{-1, 1\}$, and $\|\mathbf{v}\| < \|\mathbf{b}_k\| = \max\{\|\mathbf{b}_i\|, \|\mathbf{b}_j\|\}$, $k \in \{i, j\}$. We want to replace $\mathbf{b}_k$ with $\mathbf{v}$. If we define $\Delta ij = \mathbf{b}_i \cdot \mathbf{b}_j$, then

$$\Delta_{kh} = \begin{cases} \mathbf{v} \cdot \mathbf{b}_h = (\mathbf{b}_i + \epsilon \mathbf{b}_j) \cdot \mathbf{b}_h = \Delta_{ih} + \epsilon \Delta_{jh} & \text{if } h \neq k \\ \mathbf{v} \cdot \mathbf{v} = (\mathbf{b}_i + \epsilon \mathbf{b}_j) \cdot (\mathbf{b}_i + \epsilon \mathbf{b}_j) = \Delta_{ii} + \Delta_{jj} + 2\epsilon \Delta_{ij} & \text{if } h = k \end{cases} \tag{4.6}$$

Pre-computing the entries of Δ and storing them in an array allows for faster computation, as (4.6) can be computed more efficiently than the actual dot product when updates are needed.

The goal of this work was to find new applications of LLL and WEIGHTREDUCTION to combinatorial search problems. Before discussing our results, the next section discusses the applications basis reduction is already well-known for.

Algorithm 4.2.2: WEIGHTREDUCTION$(B = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n], \Delta)$

```
s ← 0
for i ← 1 to n − 1
do { for j ← i + 1 to n
     do { for each e ∈ {−1, 1}
          do { if Δ_jj ≥ Δ_ii  then k ← j  else k ← i
               v ← b_i + e b_j
               if ‖v‖² < Δ_kk
               then { b_k ← v and Δ_ii ← Δ_jj + 2eΔ_ij
                      s ← s + 1
                      for h ← 1 to n
                        do if h ≠ i and h ≠ j
                        then Δ_kh ← Δ_ih + eΔ_jh and Δ_hk ← Δ_kh
                      if k = i  then x ← j  else x ← i
                      Δ_kx ← Δ_ix + eΔ_jx and Δ_xk ← Δ_kx
return (B, Δ, s)
```

4.3 Past Applications

4.3.1 Integer Programming with Fixed Dimension

One of the first applications of the LLL Algorithm was that of Lenstra's in 1983 [100], where basis reduction was used to show that integer programming with fixed dimension n is solvable in polynomial time. While even Lenstra admits that the practical value of the algorithm is "restricted to small values of n" [100], his result had a large theoretical impact, which included the novel use of geometry of numbers in optimization ([115], ch. 9). In this section, we give an overview of the algorithm, with emphasis on the importance of LLL to its success.

The integer programming problem considered is the IP *feasibility problem*, which asks to decide whether a given closed convex set $K = \{\mathbf{x} \in \mathbb{R}^n \mid A\mathbf{x} \leq \mathbf{b}\}$ contains a vector in $\mathbb{Z}^n$, that is, whether

$$K \cap \mathbb{Z}^n \neq \emptyset. \tag{4.7}$$

The main idea of Lenstra's algorithm is to apply a linear transformation τ to K so that τK is "spherical" in shape. This transformation produces the lattice $\mathcal{L} = \tau\mathbb{Z}^n$, and the decision of (4.7) becomes that of deciding $\tau K \cap \mathcal{L} \neq \emptyset$. Let B be the reduced basis of $\mathcal{L}$

obtained from LLL. The structure and properties of τK, $\mathcal{L}$, and B will provide enough insight to bound the running-time by a polynomial times a constant that depends only on n. Define *ball* $B(\mathbf{p}, r)$, with center $\mathbf{p}$ and radius r, as $B(\mathbf{p}, r) = \{\mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{x} - \mathbf{p}\| \leq r\}$. We pick τ so that $B(\mathbf{p}, 1) \subseteq \tau K \subseteq B(\mathbf{p}, R)$, where R is bounded by a constant c_1 that depends only on n. See Section 2 of [100] for an effective way to do this in polynomial time.

The analysis of this algorithm makes use of the following lemma.

Lemma 4 (Lenstra 1983 [100])**.** *Let $B = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n]$ be some basis of lattice $\mathcal{L}$. For every $\mathbf{x} \in \mathbb{R}^n$, there exists some $\mathbf{y} \in \mathcal{L}$ such that*

$$\|\mathbf{x} - \mathbf{y}\|^2 \leq \frac{1}{4}\left(\|\mathbf{b}_1\|^2 + \cdots + \|\mathbf{b}_n\|^2\right). \tag{4.8}$$

We assume B is ordered such that $\|\mathbf{b}_n\| = \max_j \|\mathbf{b}_j\|$. Then, (4.8) implies that for every $\mathbf{x} \in \mathbb{R}^n$, there exists some $\mathbf{y} \in \mathcal{L}$ such that $\|\mathbf{x} - \mathbf{y}\| \leq \frac{1}{2}\sqrt{n}\|\mathbf{b}_n\|$. We set $\mathbf{p} = \mathbf{x}$ and find some $\mathbf{y} \in \mathcal{L}$ such that $\|\mathbf{p} - \mathbf{y}\| \leq \frac{1}{2}\sqrt{n}\|\mathbf{b}_n\|$. If $\mathbf{y} \in \tau K$, then we have determined $\tau K \cap \mathcal{L} \neq \emptyset$ and we stop. However, if $\mathbf{y} \notin \tau K$, further analysis is needed. Note that $\mathbf{y} \notin \tau K$ implies $\mathbf{y} \notin B(\mathbf{p}, 1)$, which in turn implies $\|\mathbf{p} - \mathbf{y}\| > 1$ and

$$\frac{2}{\sqrt{n}} < \|\mathbf{b}_n\|. \tag{4.9}$$

Let $\mathcal{L}' = \text{span}_{\mathbb{Z}}(\mathbf{b}_1, \ldots, \mathbf{b}_{n-1})$ and $H = \text{span}_{\mathbb{R}}(\mathbf{b}_1, \ldots, \mathbf{b}_{n-1})$. That is, $\mathcal{L}'$ is the lattice of the hyperplane H in $\mathbb{R}^n$. Pick $\mathbf{v} \in \mathbb{R}^n$ to be a vector orthogonal to H and let $\mathbf{h} = \text{proj}_{\mathbf{v}}(\mathbf{b}_n)$; $\|\mathbf{h}\|$ is the *distance* between $\mathbf{b}_n$ and H. Clearly,

$$\text{vol}(\mathcal{L}) = \|\mathbf{h}\| \, \text{vol}(\mathcal{L}'). \tag{4.10}$$

Recall from Hadamard's inequality that $\text{vol}(\mathcal{L}) \leq \text{wt}(B)$ and from Corollary 13.1 that $\text{wt}(B) \leq 2^{n(n-1)/4}\text{vol}(\mathcal{L})$. Combining these with (4.10) gives

$$\text{wt}(B) \leq \; 2^{\frac{n(n-1)}{4}}\text{vol}(\mathcal{L}) = \; 2^{\frac{n(n-1)}{4}}\|\mathbf{h}\| \, \text{vol}(\mathcal{L}') \; \leq \; 2^{\frac{n(n-1)}{4}}\|\mathbf{h}\| \prod_{i=1}^{n-1}\|\mathbf{b}_i\|.$$

Since $\|\mathbf{h}\| \leq \|\mathbf{b}_n\|$, we obtain

$$2^{-\frac{n(n-1)}{4}}\|\mathbf{b}_n\| \leq \|\mathbf{h}\| \leq \|\mathbf{b}_n\|. \tag{4.11}$$

Note that $\mathcal{L} = \bigcup_{i\in\mathbb{Z}} (\mathcal{L}' + i\mathbf{b}_n)$, which is strictly contained in $\bigcup_{i\in\mathbb{Z}} (H + i\mathbf{b}_n)$. That is, $\mathcal{L}$ is contained in the union of an infinite number of parallel hyperplanes of dimension $n-1$, each adjacent pair of which are at distance $\|\mathbf{h}\|$ apart. From (4.9) and (4.11) we obtain

$$\|\mathbf{h}\| > \frac{2}{\sqrt{n}}\, 2^{-\frac{n(n-1)}{4}} = c_2. \tag{4.12}$$

Therefore, the number of hyperplanes $H + i\mathbf{b}_n$ which intersect $B(\mathbf{p}, R)$ is less than $2R/c_2 \leq 2c_1/c_2$, a constant depending only on n. We fix i for each intersection, which results in $2c_1/c_2$ integer programming problems with dimension $n-1$. Each problem is then processed recursively until the base case $n = 1$ is obtained. As each instance is processed in time polynomial of the input size, the entire algorithm's running time is bounded by a polynomial of the input size times a constant that depends only on n.

4.3.2 Combinatorial Searches

The main application of LLL studied in this work is to combinatorial search problems, an application that, unlike the previous, is significant due to its success in practice. The general idea of this process is to represent a problem in a particular matrix form, so that if the matrix is treated as a basis of a lattice, the solution to the problem is found in a short vector of a reduced basis.

Many combinatorial search problems can be formulated as a system of linear equations,

$$A\mathbf{x} = Y, \tag{4.13}$$

where $\mathbf{x} \in \{0,1\}^n$ is unknown [93]. An equivalent version of (4.13) is

$$\begin{bmatrix} I & \mathbf{0} \\ A & -Y \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{1} \end{bmatrix} = \begin{bmatrix} \mathbf{x} \\ \mathbf{0} \end{bmatrix}. \tag{4.14}$$

Clearly, $\mathbf{x}$ is a solution of (4.13) if and only if it is a solution to (4.14).

If we let

$$B = \begin{bmatrix} I & \mathbf{0} \\ A & -Y \end{bmatrix} \tag{4.15}$$

be a basis of lattice $\mathcal{L}$, then the hope is that a reduced basis of $\mathcal{L}$ obtained from LLL will

contain the presumably short vector $[\ \mathbf{x},\ \mathbf{0}\ ]^T$. This method was pioneered by Lagarias and Odlyzko in 1985 [94] to attack instances of SUBSET-SUM, a problem we discuss below. Around that same time, Brickell [11] devised several different methods that also used LLL and basis reduction to attack SUBSET-SUM. A few years later in 1988, Kreher and Radziszowski [122] improved the methods of Lagarias and Odlyzko in part by introducing the WeightReduction algorithm previously described.

We now present SUBSET-SUM, as well as combinatorial design searches, and discuss the ways basis reduction has been used to attack instances of them.

SUBSET-SUM

The classical SUBSET-SUM problem asks for a subset S of a given set of integers A such that the sum of all $a \in S$ equals a given value (see for example [33]). If considering (4.13), SUBSET-SUM can be formulated as: Given a vector of positive integers $\mathbf{a} = \{a_1, a_2, \ldots, a_n\}$ and a target sum z, find vector $\mathbf{x} \in \{0,1\}^n$ such that

$$\sum_{i=1}^{n} x_i a_i = z. \tag{4.16}$$

The decision version of SUBSET-SUM is **NP**-complete, and was one of Karp's 21 **NP**-complete problems [83] (called KNAPSACK).

A number of methods involving basis reduction have been used to solve instances of this problem [94, 122, 96], particularly those with low *density*. The density of an instance with $\mathbf{a} = [a_1, \ldots, a_n]$ is defined as

$$\partial(\mathbf{a}) = \frac{n}{\log_2(\max_j a_j)}.$$

In general, if $\partial(\mathbf{a}) > 1$, many subsets of $\mathbf{a}$ will have the same sum.

If from (4.16) and (4.13) we let $A = \mathbf{a}$ and $Y = z$, then we can apply the LLL and WeightReduction to (4.15) in the hope of finding a solution. For example, take $\mathbf{a} = [29, 82, 87, 5, 76, 74, 13, 9]$ and target sum $z = 134$. We construct basis B as:

$$B = \left[\begin{array}{cccccccc|c} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \hline 29 & 82 & 87 & 5 & 76 & 74 & 13 & 9 & -134 \end{array}\right].$$

Running LLL on B gives the reduced basis

$$B' = \left[\begin{array}{cccccccc|c} 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 2 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ -1 & -1 & -1 & 0 & 0 & -1 & 0 & -1 & 0 \\ 1 & 0 & 0 & -1 & -1 & -1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & -1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & -1 & 1 & 0 \\ \hline 0 & 0 & 0 & -1 & 1 & 0 & 0 & -1 & 0 \end{array}\right].$$

The sixth column of B' contains the solution $\mathbf{x} = [1, 0, 1, 1, 0, 0, 1, 0]$.

Combinatorial Designs

In design theory, a t-(v, k, λ) design is the pair (X, B) where $X = \{1, 2, \ldots, v\}$ is a set of *points* and B is a collection of points of size k, called *blocks*, such that any t points of X are in exactly λ blocks of B (see for example [15]). A well-known 2-$(7, 3, 1)$ design is the Fano plane from Figure 3.2.

For a t-design (X, B), permuting the elements of X in such a way that preserves B is called an automorphism. The automorphism group $G \subseteq \mathrm{Sym}(X)$, where $\mathrm{Sym}(X)$ is the symmetric group containing all permutations of X, is such that every $g \in G$ is an automorphism. A group *action* of G on X consists of all permutations of X using the elements of G. An orbit of $x \in X$ is the set of elements of X that x can reach by the action of G and is written as Gx:

$$Gx = \{g \cdot x \mid g \in G\}.$$

A theorem by Kramer and Mesner [90] states that a t-(v, k, λ) design exits if and only if there is a $(0, 1)$-solution U to

$$A_{tk}U = \lambda J \tag{4.17}$$

where $J = [1, 1, \ldots, 1]^T$ and A_{tk} is defined as follows. The rows of A_{tk} are labeled by Gy for all t-subsets $y \subset X$ and the columns are labeled by Gz for all k-subsets $z \subset X$. Each entry $A_{tk}[\alpha, \beta]$ is the number of ways a representative $T \in \alpha$ is a subset of $K \in \beta$.

The basis reduction technique previously described can be applied [92, 93] by defining basis B as:

$$B = \begin{bmatrix} I & \mathbf{0} \\ A_{tk} & -\lambda J \end{bmatrix}.$$

A number of t-designs were discovered using this method [92, 91].

4.4 Graph Domination with Basis Reduction

4.4.1 Introduction

For a graph G, $D \subseteq V(G)$ is a *dominating set* of G if every vertex of G is either in D or adjacent to a vertex in D. A vertex u *dominates* vertex v if $u = v$ or $\{u, v\} \in E(G)$. The minimum cardinality of such a set is the *domination number* and is denoted $\gamma(G)$.

The origins of graph domination date back to 1862 when de Jaenisch [35] studied the mathematics of chess, and in particular, the problem of determining how many queens are needed in order to "capture" any position of an $n \times n$ chessboard. If G is a graph with n^2 vertices, each representing a position of the board, and edges defined as $\{u, v\} \in E(G)$ if and only if a queen on position u can capture a piece on position v, then this problem is equivalent to determining $\gamma(G)$. Analogous graphs and problems have been studied for a variety of other chess pieces, most notably rooks (see Figure 4.7) and knights.

The concept of domination as applied to graphs was first introduced by Berge in 1958 [7] under the term *coefficient of external stability*. In 1962, Ore [116] coined the terms *dominating set* and *domination number*. Since then, domination has grown into a substantial research area in graph theory. The standard reference text, "Fundamentals of Domination in Graphs", authored by Haynes, Hedetniemi, and Slater[71], was published nearly 15 years ago and contains over 1200 citations of work on or related to the area.

Perhaps a reason for this "explosive growth" [71] of research in this field is the numerous applications of it to real-world problems. In addition to the chess piece coverings, domination can be seen in a variety of natural problems. In [71], a number of such applications are suggested, including problems of radio station coverage, social networking, and land surveying.

In this section, we formulate the graph domination problem, MIN-DOMINATING-SET, in such a way that attacks via basis reduction can solve particular instances of it. A discussion of some related graph properties associated with graph domination follows.

4.4.2 Related Parameters and Problems

Graph domination is related to a number of other graph parameters. D is an *independent dominating set* if it is both independent and dominating. For $X \subset V(G)$ and $v \in X$, the *private neighborhood* $PN(v, X)$ is the set of vertices which are in $N_G[v]$ but not in $N_G[X \setminus x]$. D is a minimal dominating set if and only if the private neighborhood of each $v \in D$ is nonempty. $S \subset V(G)$ is an *irredundant* set if each $v \in S$ is either isolated from all other vertices in S or is adjacent to a vertex that is isolated from S, that is, $PN(v, S)$ is nonempty for all $v \in S$. Irredundance can be seen as a generalization of independence.

Proposition 1. *Every maximal independent set is a minimal dominating set.*

Proposition 2. *A set is minimal dominating if and only if it is both dominating and irredundant.*

The converse of Proposition 1 holds only when the minimal dominating set is independent, and therefore a set is a maximal independent set if and only if it is an independent dominating set.

Define:

- $ir(G)$ and $IR(G)$ as the smallest and largest cardinalities of maximal irredundant sets in G, respectively
- $i(G)$ as the smallest cardinality of a minimal independent dominating set in G
- $\Gamma(G)$ as the largest cardinality of a minimal dominating set in G

Theorem 14 presents the *domination chain*, a well-known inequality chain of parameters related to domination.

Theorem 14 (Cockayne, Hedetniemi, Miller 1978 [27])**.** *For every graph G,*

$$ir(G) \leq \gamma(G) \leq i(G) \leq \alpha(G) \leq \Gamma(G) \leq IR(G). \tag{4.18}$$

We give examples of witnesses to the strict inequalities of (4.18) in Figure 4.2. Explanations of these examples follow.

(a) $ir(G) < \gamma(G)$

This graph is known as the A-L graph and was originally given by Alan and Laskar [2]. Vertices a_1 and a_2 form a maximal irredundant set, giving $ir(G) = 2$, while at least three vertices are needed for a dominating set.

(b) $\gamma(G) < i(G)$

Vertices b_1 and b_2 form a dominating set and give $\gamma(G) = 2$. However, every maximal independent set has cardinality 3.

(c) $i(G) < \alpha(G)$

The graph has maximal independent sets of cardinalities 1 (the smallest) and 2 (the largest).

(d) $\alpha(G) < \Gamma(G)$

Either the outside or inside triangle form a minimal dominating set, giving $\Gamma(G) = 3$, while $\alpha(G) = 2$.

(e) $\Gamma(G) < IR(G)$

Originally presented in [77], this graph has $\Gamma(G) = 2$ (forced by the two vertices of degree 3) and $IR(G) = 3$ (take three vertices with degree 4 of one of the K_4's).

The domination chain has led to a wide range of research (see [71]). One popular research question asks: Does there exist a graph whose domination chain is the integer sequence $1 \leq a \leq b \leq c \leq d \leq e \leq f$ (that is, $ir(G) = a, \gamma(G) = b, \ldots, IR(G) = f$)? If such a graph exists, then $a, \ldots, f$ is a *domination sequence*. A complete classification of domination sequences can be found in [28].

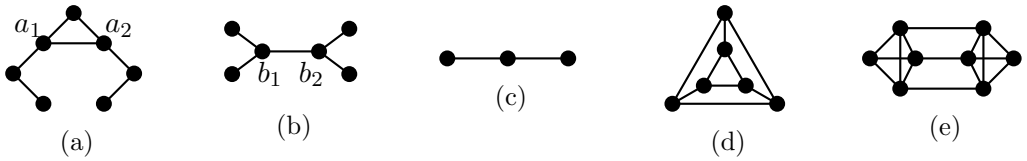


Figure 4.2: Witnesses to the strict inequalities of the domination chain.

In addition to the domination chain, many other domination-related parameters have been proposed and studied, with [71] citing more than 75 types and variations. For a graph G, a *connected dominating set* is a dominating set that induces a connected subgraph of G. A *total dominating set* is a dominating set D where each $v \in D$ is dominated by some $u \in D$, $u \neq v$. That is, $N_G(v) \cap D \neq \emptyset$ for all $v \in V(G)$. The minimum cardinalities for the connected and total dominating sets are denoted $gc(G)$ and $gt(G)$, respectively. A partition of $V(G)$ into disjoint dominating sets is called a *domatic partition*, and the maximum number of parts in a domatic partition of a graph is the graph's *domatic number* [26]. The domatic number is related to dominating sets similarly to the way the chromatic number is related to independent sets.

As irredundant sets can be seen as generalized independent sets, it is somewhat natural to ask Ramsey-type questions about them. The *irredundant Ramsey number* $s(k, l)$, originally studied by Brewster, Cockayne, and Mynhardt in 1989 [10], is the smallest n such that every edge two-coloring of K_n necessarily contains an irredundant set of order k or l in the first or second color, respectively. As every independent set is an irredundant set, it is clear that $s(k, l)$ exists for all positive k and l, and further more that $s(k, l) \leq R(k, l)$. The two smallest open cases are $18 \leq s(3, 8) \leq 22$ and $13 \leq s(4, 5) \leq 25$. For more information see for example [24].

The computational aspects of domination and its related graph parameters have received much academic attention, with over 200 papers published in the area [71]. The majority of problems associated with domination are difficult to compute, and are therefore of interest to this work. The first complexity results related to domination appeared in Garey and Johnson's classic text from 1979 [54], where the decision versions of the domination number, independent domination number, and domatic number were shown to be **NP**-complete. In

1984, Pfaff [118] showed that the decision versions of the irredundance number and connected and total domination numbers for a general graph are **NP**-complete, and remain so when the graph is bipartite [119].

4.4.3 Domination via Basis Reduction

Let A be the adjacency matrix of a graph G and let $N = A + I$ be the *closed neighborhood matrix* of G. Let $\mathbf{x} \in \{0,1\}^n$ represent a set $S \in V(G)$ with entries $x_i = 1$ if $i \in S$ and $x_i = 0$ otherwise. The domination number of G can be formulated as the following integer program (see Chapter 1 of [70]):

$$\begin{aligned} \text{Minimize} \quad & \sum_{i=1}^{n} x_i \\ \text{subject to:} \quad & N\mathbf{x} \geq \mathbf{1} \\ & \mathbf{x} \in \{0,1\}^n. \end{aligned} \tag{4.19}$$

Note that replacing N with A produces an IP for *total* domination.

The similarity between the constraints of (4.19) and the matrix equation (4.13) was our motivation for attacking graph domination problems via basis reduction. However, as the constraints are inequalities, they cannot be converted to a lattice basis that is similar to (4.15). Simply changing the constraint to the equality $N\mathbf{x} = \mathbf{1}$ asks for an *independent* dominating set D where each vertex is dominated by exactly one vertex, that is, $|N_G[v] \cap D| = 1$ for all $v \in V(G)$. In coding theory, if such a D exists it is considered a *perfect code* and G is considered a *perfect graph* (see e.g. [8]). Section 4.4.5 further discusses the relationship between graph domination and coding theory.

We follow the standard method for converting inequality constraints into equalities, by subtracting a slack variable x_{i+n} from each constraint $1 \leq i \leq n$. In matrix form this is represented with a $n \times 2n$ matrix N_s:

$$N_s = [\, N,\ -1 \cdot I \,].$$

The IP becomes:

$$\begin{aligned} \text{Minimize} \quad & \sum_{i=1}^{n} x_i \\ \text{subject to:} \quad & N_s\mathbf{x} = \mathbf{1} \\ & x_i \in \{0,1\} \quad \text{for } i = 1,\ldots,n, \\ & x_i \geq 0 \quad \text{for } i = n+1,\ldots,2n. \end{aligned} \tag{4.20}$$

We can now create a basis B from the constraints of (4.20). If $\mathbf{x}$ is the solution vector to 4.15, the hope is that $[\,\mathbf{x}\,,\mathbf{0}\,]^T \in \mathcal{L}(B)$ is a relatively short vector and can therefore be found by applying some combination of LLL and WeightReduction to B. The first basis attempted was of the same form as previously described:

$$B = \begin{bmatrix} I & \mathbf{0} \\ N_s & -\mathbf{1} \end{bmatrix}.$$

However, initial tests showed LLL and WeightReduction failing to find $\mathbf{d} = [\,\mathbf{x}\,,\mathbf{0}\,]^T$. A potential issue was that B, being made up of only 1's and 0's, had short enough vectors that little processing is needed before it is reduced. A simple, effective fix was scaling the domination IP portion of B by some positive integer c:

$$B_c = \begin{bmatrix} I & \mathbf{0} \\ cN_s & -c\mathbf{1} \end{bmatrix}. \tag{4.21}$$

Many experiments were successful by just setting $c = 2$. A summary of additional improvements to our process follows.

4.4.4 Search Improvements

An important observation is that when a minimum dominating set D is found by solving (4.20), if $v_i \in V(G) = \{v_1,\ldots,v_n\}$ is dominated by two or more vertices in D, then the corresponding slack variable x_{n+i} will be greater than 0.

Figure 4.3 displays a graph G where both vertices of its minimum dominating set $\{v_1, v_2\}$ dominate the vertex v_3.

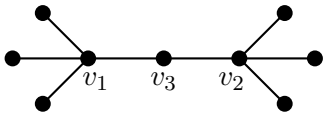


Figure 4.3: Graph with v_3 dominated by both vertices of the minimum dominating set $\{v_1, v_2\}$.

The basis (4.21) for this graph (with $c = 2$) is presented in Figure 4.4. The reduced basis obtained from applying LLL is presented in Figure 4.5, with the solution vector $\mathbf{x}$ for $\gamma(G)$ successfully appearing in the rightmost column. Note that $x_1 = 1$ and $x_2 = 1$ correspond to v_1 and v_2 being in the dominating set D, while the slack variable $x_{12} = 1$ corresponds to v_3 being dominated by D twice. All other variables $x_4, \ldots, x_9$ equal zero, as no other vertex is in D, and all other slack variables equal zero, as no other vertex is dominated by D more than once.

A plausible explanation for this success is that v_3 was dominated by two (and no more than two) vertices, and therefore $\|\mathbf{d}\|$ was relatively small. If it had been dominated by more vertices, the slack variable x_{12} would have been larger. This presents an issue when searching for $\mathbf{d} \in \mathcal{L}$, as an increase in the values of the slack variables increases the length of $\mathbf{d}$, making it less likely to appear in a reduced basis.

This situation occurs with the graph presented in Figure 4.6, where v_4 is dominated by all three vertices of the minimum dominating set $\{v_1, v_2, v_3\}$. After applying LLL, the reduced basis B_2' does not contain a vector $\mathbf{d}$ which corresponds to a dominating set of the graph. The desired vector is $\mathbf{d} = [1, 1, 1, 0, ..., 0, 2, 0, ..., 0]$, that is,

$$d_i = \begin{cases} 1 & \text{if } i = 1, 2, 3, \\ 2 & \text{if } i = 17, \\ 0 & \text{otherwise.} \end{cases}$$

Although $\mathbf{d}$ is not in B_2', the two rightmost column vectors $\mathbf{d}_1$ and $\mathbf{d}_2$ of B_2' are such that $\mathbf{d}$ is a linear combination of the two, that is, $\mathbf{d} = \mathbf{d}_1 + \mathbf{d}_2$, where

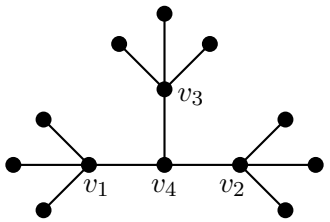
$$\mathbf{d}_1 = [\,0\ \ 0\ \ 1\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ -1\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 1\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ \ 1\ \ \ 1\ \ 0 \ldots 0\,]^T,$$
$$\mathbf{d}_2 = [\,1\ \ 1\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ \ 1\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 1\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ -1\ -1\ \ 0 \ldots 0\,]^T.$$

$$\left[\begin{array}{cccccccccccccccccc|c}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
\hline
2 & 0 & 2 & 2 & 2 & 2 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 \\
0 & 2 & 2 & 0 & 0 & 0 & 2 & 2 & 2 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 \\
2 & 2 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & -2 \\
2 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & -2 \\
2 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 0 & -2 \\
2 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & -2 \\
2 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & 0 & 0 & -2 \\
0 & 2 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & 0 & -2 \\
0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & -2
\end{array}\right]$$

Figure 4.4: The lattice basis for finding dominating sets of Figure 4.3

$$\left[\begin{array}{cccccccccccccccccc|c}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & \mathbf{1} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & \mathbf{1} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \mathbf{0} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & \mathbf{0} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{0} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{0} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & \mathbf{0} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & \mathbf{0} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \mathbf{0} \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & \mathbf{0} \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & \mathbf{0} \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \mathbf{1} \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{0} \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \mathbf{0} \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & \mathbf{0} \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & \mathbf{0} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & \mathbf{0} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & \mathbf{0} \\
-2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{0} \\
0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{0} \\
0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{0} \\
0 & 0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{0} \\
0 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{0} \\
0 & 0 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{0} \\
0 & 0 & 0 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{0} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{0} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{0}
\end{array}\right]$$

Figure 4.5: The reduced basis of Figure 4.4 found from running LLL. The solution vector for γ is the rightmost column.



	v_1	v_2	v_3	v_4									
v_1	0	0	0	1	1	1	1	0	0	0	0	0	0
v_2	0	0	0	1	0	0	0	1	1	1	0	0	0
v_3	0	0	0	1	0	0	0	0	0	0	1	1	1
v_4	1	1	1	0	0	0	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	0	0	0	0	0	0
	0	1	0	0	0	0	0	0	0	0	0	0	0
	0	1	0	0	0	0	0	0	0	0	0	0	0
	0	1	0	0	0	0	0	0	0	0	0	0	0
	0	0	1	0	0	0	0	0	0	0	0	0	0
	0	0	1	0	0	0	0	0	0	0	0	0	0
	0	0	1	0	0	0	0	0	0	0	0	0	0

Figure 4.6: Graph with v_4 dominated by all three vertices of the minimum dominating set $\{v_1, v_2, v_3\}$.

Note that $\|\mathbf{d}\|^2 = 7$ while $\|\mathbf{d}_1\|^2 = 5$ and $\|\mathbf{d}_2\|^2 = 6$. The solution $\mathbf{d}$ was not in the basis because both $\mathbf{d}_1$ and $\mathbf{d}_2$ were shorter.

This observation led to a number of modifications of our search. The first involved considering sums and differences of the vectors of the basis after it had been processed by LLL and WeightReduction . Experiments showed that considering similar linear combinations of three vectors also found previously undiscovered dominating sets, and therefore were also considered. These checks proved insignificant to the overall running time of the computations, both in theory, with only an additional $O(2n^2 + 3n^3)$, and in practice. In Section 4.4.6 we refer to these checks in rounds, namely, that rounds 1, 2, and 3 correspond to checks with one column vector, two columns added and subtracted, and three columns added and subtracted, respectively.

Different Distance Metric

The second modification attacked the issue presented above more directly: the metric used to measure a vector's "length." In addition to the standard Euclidean norm, tests were done using the L_1 norm, also known as the *taxicab* norm, as defined as

$$\|\mathbf{x}\|_1 = \sum_{i=1}^{n} |x_i|.$$

Under this metric, the vector entries are not squared, and there is therefore less of a "punishment" for having entries greater than 1 in a vector of a short basis. In the above example, the length of $\mathbf{d}$ is decreased to $\|\mathbf{d}\|_1 = 5$, while the lengths of $\mathbf{d}_1$ and $\mathbf{d}_2$ remain the same at 5 and 6, respectively. If WeightReduction was applied to this basis under

this new norm, $\mathbf{d}$ would have replaced $\mathbf{d}_2$ and the solution would have been found.

During our experiments, combinations of the two norms were applied to LLL and WeightReduction . Specifically, three cases were considered:

1. LLL and WeightReduction both under the Euclidean norm
2. LLL under Euclidean and WeightReduction under L_1
3. Both under L_1

In general, better results were found with 1 and 2, while 3 was the computationally fastest.

Converting to 0-1 Integer Program

We tested another version of the IP (4.20) which allowed for the entire unknown $\mathbf{x}$ to be a 0-1 vector. Results obtained with this formulation improved over those of (4.20), as the solution vector was shorter in general, and therefore appeared more often in reduced bases. Our main conversion was a replacement of each slack variable x_{i+n} with a sequence of variables that represented x_{i+1} in binary. That is, we replaced entry $n_{i,i+1} = -1$ of N_S with entries $[-1, -2, \ldots, -2^l]$ so that x_{i+n} could be found as a bitvector instead of a positive integer.

It was necessary to bound l by a reasonable number so that our basis did not contain too many vectors. Recall that the purpose of the slack variables is to catch the case when a vertex v is dominated by more than vertex in the dominating set. A somewhat trivial upper bound on the number of such vertices is the degree of v. Using this bound, each x_v required $\lfloor \log_2 \deg(v) \rfloor + 1$ binary slack variables $[-1, -2, \ldots, -(2^{\log_2 \deg(v)})]$. The total number of slack variables β was then

$$\beta = |V(G)| + \sum_{v \in V(G)} \lfloor \log_2 \deg_G(v) \rfloor .$$

Let N_β be the $n \times (n + \beta)$ matrix created this way. Our final IP is:

$$\begin{aligned} \text{Minimize} \quad & \sum_{i=1}^{n} x_i \\ \text{subject to:} \quad & N_\beta \, \mathbf{x} \geq \mathbf{1} \\ & \mathbf{x} \in \{0,1\}^{n+\beta}. \end{aligned} \tag{4.22}$$

There were a number of benefits to this bounding approach. An obvious one was that vertices with low degree did not have any unnecessary slack, and as the number of the variables grew logarithmically, the total β never became too large that the basis could not be processed. Another benefit was that if β was relatively large (especially when compared to the size of basis (4.21)), it generally implied that the average degree of G was large. We will discuss in the next section how dominating sets of graphs with larger and irregular degrees were usually easier to find. In other words, as β became larger, the basis became easier to process. Therefore, there was rarely a disadvantage to using this method over the previous.

Varying Parameters on Input

In addition to these variations, changes to the other parameters of our process were investigated. Most notably was the parameter y of the definition of a y-reduced basis. While setting $y = \frac{3}{4}$ is what was originally considered in [99], Algorithm 4.2.1, and Theorem 4.3, any y strictly between $\frac{1}{4}$ and 1 will preserve the polynomial-time complexity of LLL. In general, y is used as a parameter for "how reduced" the basis is, with a higher y leading to a stronger reduction. Throughout testing, a number of different y values in $[\frac{3}{4}, 1)$ were tested, with higher numbers often leading to better results. Another parameter, the scale c of B_c (4.21), was increased from 2 to other values, most notably 10 and 100.

Further Modifications to LLL **and** WeightReduction

The final modifications to our process involved a change to Algorithm 4.2.1 to improve the overall experimental running time. The main bottleneck of the algorithm, when implemented, was the frequency of calls to Gram-Schmidt. To fix this, we simply did not call it as often. Step 2 of the algorithm was changed so that all vectors $\mathbf{b}_j^*$, $\mathbf{b}_{j+1}^*$ were checked, instead of just a few, before returning to Step 1. This modification is presented in Algorithm 4.4.1. Although a number of other modifications were attempted, this one proved to have the best balance between running time and reduction performance.

A variety of combinations of LLL2 and WeightReduction were considered. The first combination was simple; it called LLL2 once, followed by WeightReduction as many times as possible until the weight was no longer reduced. This is presented as Algorithm 4.4.2. A more successful combination was that of Radziszowski and Kreher [122]. Algorithm

Algorithm 4.4.1: LLL2($B = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n], y$)

Step 1
$([\mathbf{b}_1^*, \mathbf{b}_2^*, \ldots, \mathbf{b}_n^*], \{\mu_{j,i}|_{1 \le i < j \le n}\}) \leftarrow$ GRAM-SCHMIDT$(\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n)$
for $j \leftarrow 2$ **to** n
  do { **for** $i \leftarrow j-1$ **downto** 1
    do { **if** $|\mu_{ij}| > 1/2$
      then $\mathbf{b}_j \leftarrow \mathbf{b}_j - \lfloor \mu_{i,j} + \frac{1}{2} \rfloor \mathbf{b}_i$

Step 2
$([\mathbf{b}_1^*, \mathbf{b}_2^*, \ldots, \mathbf{b}_n^*], \{\mu_{i,j}|_{1 \le i < j \le n}\}) \leftarrow$ GRAM-SCHMIDT$(\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n)$
$found =$ **false**
for $j \leftarrow 1$ **to** $n-1$
  do { **if** $\|\mathbf{b}_{j+1}^* + \mu_{j+1,j}\mathbf{b}_j^*\|^2 < \frac{3}{4}\|\mathbf{b}_j^*\|^2$
    then { swap $\mathbf{b}_j$ and $\mathbf{b}_{j+1}$
      $found = true$
if $found$
  then go to Step 1

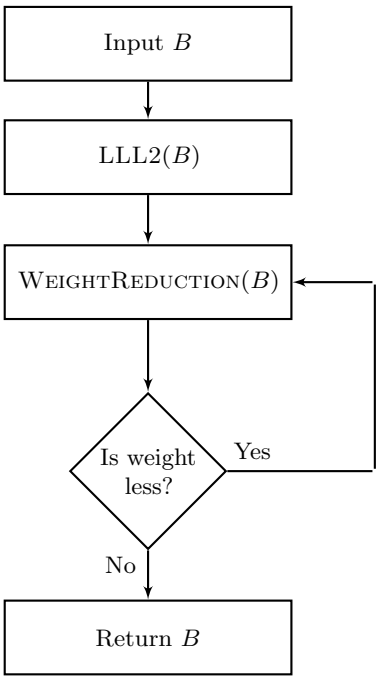
return $([\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n])$

4.4.3 displays this process, where SORT(B) sorts the basis vectors from shortest to longest, forcing the shortest vector to always be in the first column.

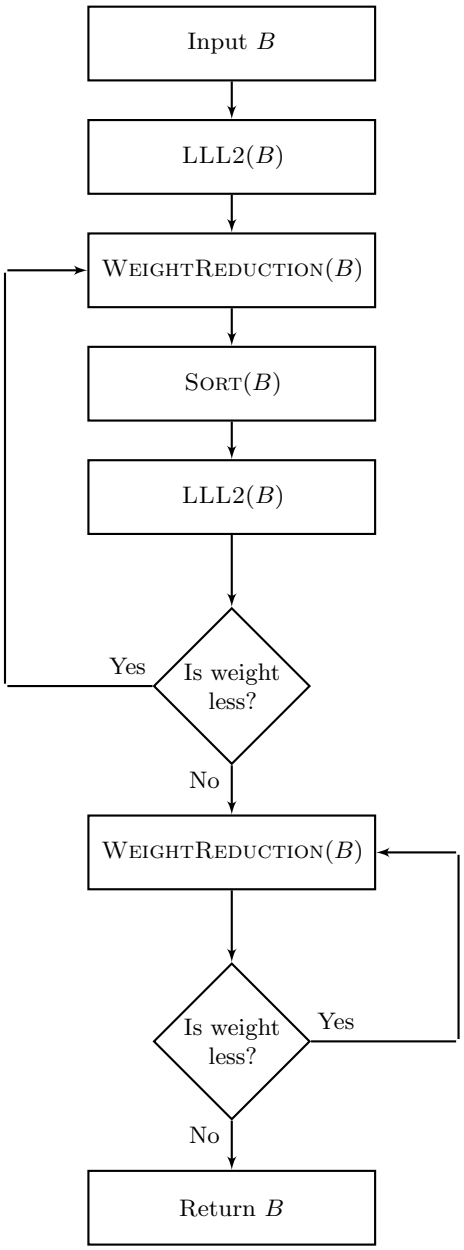
4.4.5 Football Pool Problem

Before presenting our experimental results, we introduce a domination problem regularly associated with coding theory, as our experiments include attacks on specific instances of it. The *football pool problem* involves the popular recreation of guessing (or betting on) the outcomes of a number of competitions [68]. These competitions are classically regarded as football matches. The challenge is to determine the minimum number of guesses one needs to make in order to guarantee that a desired prize is won, regardless of the results. In many cases, instances of the problem depend on few parameters, seem natural, and can be easily explained. Despite this apparent simplicity, they often involve sophisticated combinatorial structures and are very difficult to solve.

Algorithm 4.4.2: REG-REDUCTION($B = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n]$)



Algorithm 4.4.3: RK-REDUCTION($B = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n]$)



The classical football pool operates as follows. In general, a football match has three possible outcomes: win, lose, or tie. Let n be the number of matches considered in the pool; the sequence of outcomes of these matches is encoded in a ternary vector $\mathbf{v} \in \mathbb{Z}_3^n$. Before $\mathbf{v}$ is determined, participants place one or more forecasts in the pool, where each forecast $\mathbf{g} \in \mathbb{Z}_3^n$ contains a sequence of outcome guesses. A participant receives the top prize if they guess each outcome successfully, that is, if $\mathbf{g} = \mathbf{v}$ for some forecast $\mathbf{g}$. Define $d(\mathbf{g}, \mathbf{v})$ as the number of incorrect guesses of $\mathbf{v}$ in $\mathbf{g}$, and let R be the maximum $d(\mathbf{g}, \mathbf{v})$ a participant can place and still receive a prize. The classical football pool problem asks to find the minimum number of forecasts needed in order to win a prize when $R = 1$, that is, when at most one guess is allowed to be incorrect.

In coding theory, finding such minimal forecast systems is equivalent to finding minimal sized covering codes (see [29]). We now restate the football pool problem using the language of coding theory. Given *words* $\mathbf{x}, \mathbf{y} \in \mathbb{Z}_q^n$, where $\mathbf{x} = x_1 x_2 \dots x_n$ and $\mathbf{y} = y_1 y_2 \dots y_n$, the *Hamming distance* $d(\mathbf{x}, \mathbf{y})$ is defined as $| \{ i \mid x_i \neq y_i,\ i = 1, \dots, n \} |$. A *code* $C \subseteq \mathbb{Z}_q^n$ *covers* $\mathbb{Z}_q^n$ with *radius* R if for every word $\mathbf{x} \in \mathbb{Z}_q^n$ there exists some $\mathbf{c} \in C$ such that $d(\mathbf{x}, \mathbf{c}) \leq R$. The minimum cardinality of such a code is denoted $K_q(n, R)$.

An interesting aspect of the search for covering codes is that many have been discovered by pool enthusiasts rather than mathematicians [6]. In 1947, one such enthusiast, Juhani Virtakallio, published a system of 749 forecasts in the Finnish football pool magazine *Veikkaaja*. It was intended for 11-match pools having $R = 2$, and remarkably, the system always contained a forecast that earned a prize. Virtakallio therefore determined $K_3(11, 2) \leq 749$ without having any knowledge of covering codes. His system was independently discovered by Golay two years later [59], and is now known as the ternary Golay code. This code, as well as the binary Golay codes, still receive much interest, as they have particularly deep connections to several areas of mathematics (see [107] for more details). In addition to Virtakallio, many other enthusiasts are responsible for the discovery of covering codes. Such results can be found in Hämäläinen and Rankinen's 1991 survey [69], where they cite results from 13 books and magazines concerning football pools.

Covering Codes as Graph Domination

With a simple graph construction, the problem of finding minimum covering codes reduces to the problem of finding minimum dominating sets. The type of graphs used in this equivalence are the *Hamming graphs*. The Hamming graph $H_{n,q}$ is defined as $V(H_{n,q}) = \mathbb{Z}_q^n$ and $E(H_{n,q}) = \{\{u, v\} \mid d(u, v) = 1\}$ and is equivalently the Cartesian product of n copies of K_q. $H_{n,q}$ has q^n vertices, chromatic number q, diameter n, is $(nq - n)$-regular, and is vertex transitive. When $n = 1$, $H_{1,q}$ is K_q; when $n = 2$, $H_{2,q}$ is the $q \times q$ lattice (or rook's) graph; and when $q = 2$, $H_{n,2}$ is the hypercube graph Q_n. For more on Hamming graphs, see for example [12]. For more on the connection between graphs and coding theory, see for example [15].

Clearly, $K_q(n, 1) = \gamma(H_{n,q})$. When attacking these problems with basis reduction, we simply convert the covering code search to this dominating set search and use the basis reduction methods previously discussed. Complications arise when attempting $K_q(n, R)$ with $R > 1$. We overcome this by generalizing the Hamming graph to another parameter. Define graph $H_{n,q,R}$ as $V(H_{n,q,R}) = \mathbb{Z}_q^n$ and $E(H_{n,q,R}) = \{\{u, v\} \mid d(u, v) \leq R\}$.

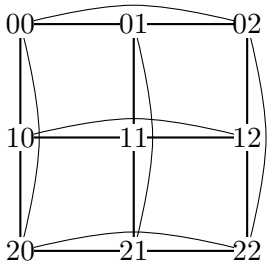


Figure 4.7: $H_{2,3}$, also known as the rook's graph or the lattice graph $L_{3,3}$.

In 1995, Davies and Royle [34] used tabu search methods to attack graph domination, using Hamming and related graphs as benchmark test instances. Their results were successful, matching many of the known upper bounds and improving on some "mixed" cases.

Similar to Folkman numbers, determining upper bounds on $K_q(n, R)$ is existential, while determining lower bounds is universal. That is, determining $K_q(n, R) \leq s$ is achieved by producing a covering code of size s, while determining $K_q(n, R) > t$ is proving that no covering code of size t exists. In recent years, the establishment of both lower and upper bounds have made use of large computations.

The known results for values and bounds of various covering code problems follow. The three problems in the scope of this work are:

1. The *classical football pool problem*, as previously described, assumes each football match has three outcomes, and is therefore concerned with codes covering $\mathbb{Z}_3^n$. The minimal covering codes associated with this problem are $K_3(n, R)$ with particular focus placed on $R = 1$
2. The *binary covering problem* is similar to the classical problem, except for each football match, the participant is "confident" that some outcome will not occur. We therefore cover $\mathbb{Z}_2^n$ and work on $K_2(n, R)$
3. The *mixed* covering problem is a combination of classical and binary: we exclude one of the three outcomes in some, but not all, of the n matches. The problem is then to cover $\mathbb{Z}_3^{n_1} \times \mathbb{Z}_2^{n_2}$, and we introduce the notation $K_{3,2}(n_1, n_2, R)$ for the minimum such covering code.

The Classical Football Pool Problem

The classical football pool problem is that of determining $K_3(n, 1)$. Known values and bounds for varying n are found in Table 4.1 [85], with $n = 6$ being the first open case.

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$K_3(n, 1)$	1	3	5	9	27	71 73	156 186	402 486	1060 1269	2854 3645	7832 9477	21531 27702	59049	166610 177147

Table 4.1: Known values and bounds for $K_3(n, 1)$ [85].

Considering lower bounds on $K_3(n, 1)$, a covering code has at least $3^n/(2n + 1)$ words, as each word in $\mathbb{Z}_3^n$ is at distance 1 from $2n$ other words. This bound is known as the *sphere covering bound*. When n is of the form $(3^k - 1)/2$, this bound is tight, which gives $K_3(13, 1) = 59049$ as shown in Table 4.1. With general R the sphere covering bound becomes

$$K_3(n, R) \geq 3^n \Bigg/ \sum_{i=0}^{R} \binom{n}{i} 2^i \ .$$

The first open case is $71 \leq K_3(6, 1) \leq 73$ and it is of the most interest. In 2002, the bound $K_3(6, 1) \geq 65$ was obtained via computations which interestingly made use of the LLL algorithm for covering code enumeration [117]. The next and current best lower bound

of 71 was obtained by Linderoth, Margot, and Thain in 2009 [103] by attacking the integer program (4.19) with grid computing. The authors note that at that time and to their knowledge, the computations were "the largest branch-and-bound computation ever run on a wide-area grid," using more than 140 CPU years. It is believed that $K_3(6,1) = 73$.

Binary Covering Problem

As previously mentioned, the *binary covering problem* involves the case where the football pool participant is "confident" that one of the three cases will not occur, and therefore only needs to consider the remaining two. This can also be interpreted as if a "tie" is not a possible outcome of a match. The question is then to find minimum dominating sets of the Hamming graphs $H_{n,2}$, often known as the hypercube graphs Q_n. These graphs are constructed on 2^n vertices, each representing a bitstring of length n. Two vertices and connected if and only if the two corresponding bitstrings differ in one bit.

Known values and bounds for $K_2(n,1)$ are presented in Table 4.2. An analogous sphere covering bound exists for the lower bound:

$$K_2(n,R) \geq 2^n \Bigg/ \sum_{i=0}^{R} \binom{n}{i}.$$

This bound is tight if and only if $r = 1$ and $n = 2^k - 1$; $r = 3$ and $n = 23$; $n = 2r + 1$; or $n = r$ (see e.g. [107]).

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$K_2(n,1)$	1	2	2	4	7	12	16	32	62	107 120	180 192	342 380	598 704	1172 1408	2048

Table 4.2: Known values and bounds for $K_2(n,1)$ [85].

Mixed Covering Problem

The final covering code problem we consider is the *mixed covering problem*, where we are "confident" that one case will not occur in only a fraction of the matches, The problem is then to find the minimum covering code of $\mathbb{Z}_3^{n_1} \times \mathbb{Z}_2^{n_2}$ with radius R, denoted $K_{3,2}(n_1, n_2, R)$. We define the graph F_{n_1,n_2} analogously to the previously defined Hamming graph. Clearly, $F_{n_1,0} = H_{n_1,3}$ and $F_{0,n_2} = H_{n_2,2}$. Table 4.3 presents known values and bounds for $K_{3,2}(t,b,R)$

with $0 \leq t \leq 12$ and $0 \leq b \leq 12$. The generalized sphere covering bound is,

$$K_{3,2}(t,b,R) \geq 3^t 2^b \Bigg/ \sum_{i=0}^{R} \sum_{j=0}^{i} \binom{b}{j} \binom{t}{i-j} .$$

b \ t	1	2	3	4	5	6	7	8	9	10	11	12
1	2	4	9	18	45 54	113 132	293 333	772 948	2072 2520	5624 6804	15405 18954	42516 52488
2	3	6	16	36	80 96	204 252	525 648	1395 1728	3770 4752	103311 13122	28439 34992	78732 102060
3	6	12	24	60 72	148 168	386 468	1022 1296	2747 3374	7463 9450	20458 25272	54507 69984	
4	8	20	44 48	107 128	268 324	700 864	1864 2304	5047 6408	13802 17496	37792 49086		
5	16	32 36	78 92	195 238	509 624	1353 1620	3641 4374	9904 11664	26244 34992			
6	24	57 64	140 171	356 432	936 1184	2500 2916	6762 8532	18257 23328				
7	42 48	101 122	256 312	672 852	1791 1944	4827 5832	12725 15552					
8	76 84	187 232	480 576	1257 1296	3353 3888	8887 10944						
9	134 160	338 408	888 1056	2370 2592	6221 7776							
10	253 284	646 768	1689 2016	4366 5184								
11	448 548	1173 1472	3072 3456									
12	869 992	2169 2560										

Table 4.3: Known values and bounds for $K_{3,2}(t,b,1)$ [85].

4.4.6 Experiments and Results

Basic Graphs

The first graphs tested were well-known graphs with trivial minimum domination sets, such as K_n, C_n, P_n, etc. Table 4.4 displays the graphs tested and their domination numbers. Computing the domination number of these graphs proved successful. In all cases, exact determination of γ was achieved. For $n \leq 401$, this could be done in under 4 CPU hours. Table 4.5 shows the times for some such cases.

G	K_n	$\overline{K_n}$	$K_{n,m}$	C_n	P_n	W_n	$K_{1,n}$	$K_{n_1,n_2,\ldots,n_k}$
$\gamma(G)$	1	n	2	$\lceil n/3 \rceil$	$\lceil n/3 \rceil$	1	1	2

Table 4.4: Graphs and their domination numbers.

Recall from Section 4.4.3 that a *perfect code* is a dominating set such that each vertex is dominated by exactly one vertex, that is, $|N_G[v] \cap D| = 1$ for all $v \in V(G)$. Note that such a set is an independent dominating set. Clearly, all graphs of Table 4.4, except the partite graphs $K_{n,m}$ and $K_{n_1,n_2,\ldots,n_k}$, have minimum dominating sets which are perfect codes. This may be an explanation for the success of basis reduction on these graphs, as such solution vectors of the IPs (4.20) and (4.22) have all slack variables set to 0, and therefore have much shorter vectors in the basis (4.21). This turns out to be true for other sets of graphs as well, which we discuss in the following.

Random Graphs

The next set of graphs tested were the classical Erdős-Rényi random graphs $G(n, p)$. The main tests included the generation of a large number of such graphs with varying n and p. Graphs on up to 100 vertices were tested with a variety of the methods discussed. In general, as the graphs became more dense, it became easier to find dominating sets with basis reduction. One such test involved attacking 20 $G(n, p)$ graphs for each $n \in \{30, 40, \ldots, 100\}$ and $p \in \{0.1, 0.2, \ldots, 0.9\}$. For all n, whenever p was greater than or equal to 0.5, a dominating set was found. We verified that these sets were minimum using `sage` [137]. Clearly, this is also true for $p = 0$ and $p = 1$, as these graphs are $\overline{K_n}$ and K_n, previously described.

Table 4.6 presents results for $n = 30$. Note that as p increased, the dominating sets

	n	10	11	50	51	100	101	200	201	300	301	400	401
K_n	γ	1	1	1	1	1	1	1	1	1	1	1	1
	s	0.03	0.04	5.96	6.33	68.87	71.42	892.85	909.50	4151.69	4207.57	12599.00	12723.90
$\overline{K_n}$	γ	10	11	50	51	100	101	200	201	300	301	400	401
	s	0.04	0.06	6.12	6.50	68.71	71.18	892.74	909.33	4159.52	4213.20	12604.80	12724.15
C_n	γ	4	4	17	17	34	34	67	67	100	101	134	134
	s	0.02	0.03	3.33	3.61	36.83	38.16	465.24	470.34	2130.49	2170.41	6448.98	6510.63
P_n	γ	4	4	17	17	34	34	67	67	100	101	134	134
	s	0.01	0.01	1.70	1.84	24.09	23.71	334.68	339.72	1627.22	1688.75	5641.80	5880.25
$K_{1,n}$	γ	1	1	1	1	1	1	1	1	1	1	1	1
	s	0.04	0.04	6.48	6.92	72.35	75.01	918.54	935.92	4239.58	4293.13	12822.09	12934.70
W_n	γ					1	1	1	1	1	1	1	1
	s	0.04	0.05	8.98	9.40	104.63	108.07	1362.67	1385.59	5057.39	5084.73	15612.75	16011.31

Table 4.5: Times of domination numbers found for graphs with varied order (s is seconds), for $y = 0.75$, $c = 2$, Algorithm 4.4.2, L_2 norm.

were found at round 1 more frequently, and that the maximum running time increased until $p = 0.6$, where it then started decreasing. Similar patterns were found for all tested n.

Chess Graphs

As previously mentioned throughout this Chapter, graphs representing chess piece coverings on chessboards are often constructed and studied. Define $n \times m$ chess graphs for queen Q, rook R, bishop B, and knight K pieces as $V = \mathbb{Z}_n \times \mathbb{Z}_m$ and $E = \{\{u, v\} \mid$ piece on position u can capture $v\}$. All such graphs tested this way were generated with `sage` [137]. The domination number of chess graphs determines minimum number of pieces to "cover" board. Numbers are known for some generalized chess pieces, such as $\gamma(R) = \min\{m, n\}$ and $\gamma(B) = n$ when $n = m$ (see [25]).

In general, we could determine γ up to 6×6 chess pieces. We could sometimes do so with 10×10 sized graphs with special searches. However, for all graphs with sizes less than 10×10, only at most 25 out of the 45 total graphs tested. This was unfortunately still true for the Rook graphs, even though the domination number is known for general $n \times m$.

Covering Codes

The Hamming graphs $H_{n,q}$ discussed in Section 4.4.5 were studied for $q = 2$ (hypercube graphs), $q = 3$ (football pool problem), and the mixed case when coverings were over

p	Rounds			Total	Max Time (s)
	1	2	3		
0.1	1	4	14	19	26.14
0.2	0	7	34	41	52.35
0.3	2	25	68	95	89.06
0.4	2	64	34	100	116.53
0.5	20	74	6	100	136.03
0.6	52	45	3	100	176.91
0.7	71	28	1	100	153.51
0.8	75	25	0	100	145.40
0.9	92	8	0	100	108.68

Table 4.6: Basis reduction results for $\gamma(G(n,p))$, $n = 30$, and $p \in \{0.1, 0.2, \ldots, 0.6\}$

$\mathbb{Z}_2^{n_1} \times \mathbb{Z}_3^{n_2}$. As previously stated, finding perfect codes (ie. Hamming codes) for graphs of the form $n = \frac{1}{2}(3^k - 1)$ tended to be easy. On the other hand, finding dominating sets for graphs with general n proved difficult. This difference in the basis reduction's abilities is shown in Tables 4.7 and 4.8, where some results for $H_{3,3}$ and $H_{4,3}$ are presented. Columns LLL and WeightReduction (WR) correspond to which norm was used for which part of the basis reduction. The "Round" column represents which round of linear combinations the dominating set was found in. That is, Round 1 is no combinations, Round 2 is combinations of two vectors, and Round 3 is combinations of three vectors.

Unfortunately, basis reduction failed to find nontrivial dominating sets for even the graphs $H_{5,3}$ and $H_{6.2}$ (with orders 243 and 64, respectively). This led us to attempt additional search techniques, described below.

More Search Improvements?

More search improvements were attempted in order to find dominating sets when our basis reduction techniques failed to do so. When using Algorithm 4.4.3 RK-Reduction with the 0-1 IP basis (4.22), we noticed that the reduced basis often had one vector with a large length (large number of 1's) and all others had short length (small number of 1's). As an example of this, we looked at the *WorldMap Graph* outputted by `sage` [137]. This graph contains a vertex for almost all countries, where two vertices are adjacent if and only if the corresponding countries share a common boundary. This graph W has order 166 and size

LLL	WR	y	found?	γ	Round	Time (s)
L_1	L_1	0.75	N	n/a	n/a	12.8
		0.80	N	n/a	n/a	13.90
		0.85	Y	6	3	15.19
		0.90	Y	5	3	25.51
		0.95	Y	5	3	29.51
		0.99	N	n/a	n/a	35.87
L_2	L_1	0.75	Y	5	3	18.17
		0.80	Y	6	3	20.04
		0.85	N	n/a	n/a	20.42
		0.90	Y	5	3	21.22
		0.95	Y	5	3	22.93
		0.99	Y	5	3	24.73
L_2	L_2	0.75	Y	5	3	17.71
		0.80	Y	5	3	19.93
		0.85	N	n/a	n/a	20.22
		0.90	Y	5	3	20.85
		0.95	Y	5	3	22.54
		0.99	Y	5	3	24.38

Table 4.7: Results for $H_{3,3}$ with $c = 10$, Algorithm 4.4.2

LLL	WR	y	found?	γ	Round	Time (s)
L_1	L_1	0.75	Y	9	1	392.68
		0.80	Y	9	1	789.02
		0.85	Y	9	1	819.69
		0.90	Y	9	1	1144.66
		0.95	Y	9	1	1089.54
		0.99	Y	9	1	1979.39
L_2	L_1	0.75	Y	9	1	1524.88
		0.80	Y	9	1	1521.30
		0.85	Y	9	1	1537.23
		0.90	Y	9	1	1546.96
		0.95	Y	9	1	1573.33
		0.99	Y	9	1	1678.38
L_2	L_2	0.75	Y	9	1	1523.45
		0.8	Y	9	1	1520.88
		0.85	Y	9	1	1557.09
		0.90	Y	9	1	1626.50
		0.95	Y	9	1	1657.10
		0.99	Y	9	1	1765.42

Table 4.8: Results for $H_{4,3}$ with $c = 10$, Algorithm 4.4.2

323. The dominating number of W decides the minimum number of countries needed so that all countries are either picked or bordered by one that is.

It is known that $\gamma(W) = 38$. Unfortunately, this could not be determined with our basis reduction techniques. However, after basis reduction, the reduced lattice basis contained a total of 415 column vectors that were valid, that is, of the form $[\mathbf{v},\ \mathbf{0}]$. The frequency of non-zero entries corresponding to non-slack variables were: 398 with one 1, 16 with two 1's, and one with 32 1's. Clearly, the single vector $\mathbf{x}$ with 32 nonzero entries was of interest, and although the 32 corresponding vertices were not a dominating set, they did dominate 147 of the 166 total.

We ran the following greedy steps to find a dominating set D:

1. Let D be the vertices corresponding to the single, long vector $\mathbf{x}$.

2. While D is not a dominating set:
   Find the remaining column vector that increases the total number of dominated vertices the most, and add the corresponding vertices to D. Repeat.

Although this process did not find additional minimum dominating sets, it did sometimes find relatively small ones. For W, the dominating set found contained 44 vertices instead of the minimum 38. For $H_{6,2}$, the dominating set was of order 16 instead of the minimum 12.

Concluding Remarks

Although the basis reduction techniques failed to find minimum dominating sets for hard instances of Chess and Hamming graphs, it did work well for other types of graphs. Our techniques were especially successful when the dominating sets were perfect codes.

The greedy search method described above is clearly worthy of further investigation. It may be possible to use stochastic optimization techniques such as simulated annealing or genetic algorithms, with the unique long vector $\mathbf{x}$ as the initial dominating set.

Chapter 5

Conclusion and Future Work

Throughout this thesis, techniques of combinatorial computing and optimization were applied successfully to a number of problems associated with graph theory. Clearly, in all cases, further work can be done and more techniques should be considered for the problems we studied. The main goal for our future work involves a further investigation of ways lattice basis reduction, linear programming, and semidefinite programming can be used to attack problems in Ramsey theory. It is our hope that such techniques can aid in determining, for example, whether $F_e(3,3;4) < 100$.

Below, we discuss one such method that we think deserves further consideration. We construct a basis of a lattice such that triangle-free colorings of graphs represented as 0-1 vectors are part of the lattice. Perhaps some combination of basis reduction and a greedy approach using semidefinite programming can be applied to such representations to show that $G \not\to (3,3)$ for graphs G of particular interest.

One of the initial goals of this thesis was to determine if basis reduction could be used for attacking problems in Ramsey theory. Although no substantial results in this direction were obtained, there may be hope in a method similar to the ones used for graph domination.

Given graph G, we enumerate all triangles $T = \{T_1, T_2, \ldots, T_{t_\triangle(G)}\}$ and edges $E = \{e_1, e_2, \ldots, e_{|E(G)|}\}$. Let $t = |T|$ and $e = |E|$ and construct a $t \times e$ matrix $\mathcal{T}_G$ as follows:

$$\mathcal{T}_G(i,j) = \begin{cases} 1 & \text{if } e_j \in T_i, \\ 0 & \text{otherwise.} \end{cases} \tag{5.1}$$

$\mathcal{T}_G$ will have $3t$ non-zero entries. If we define matrix $\mathcal{T}$ similarly to (4.4.3),

$$\mathcal{T} = [\mathcal{T}_G,\ -1 \cdot I]\,,$$

then we obtain the following:

Lemma 5. *$G \not\to (3,3)$ if and only if there exists a vector $\mathbf{x} \in \{0,1\}^e$ such that $\mathcal{T}\mathbf{x} = \mathbf{1}$.*

Proof. In order for $G \not\to (3,3)$, each triangle $T_i = \{e_{i_1}, e_{i_2}, e_{i_3}\}$ must contain two edges of one color and one edge of the other. Let $\mathcal{T}_i$ be the i-th row of $\mathcal{T}$ and x_i be the slack variable of the row. Then T_i is non-monochromatic if and only if $T_i \cdot \mathbf{x} = 1$. This is possible through either of two possibilities (without loss of generality):

1. $\mathbf{x}[e_{i_1}] = 1$, $\mathbf{x}[e_{i_2}] = \mathbf{x}[e_{i_3}] = 0$, and $x_i = 0$
2. $\mathbf{x}[e_{i_1}] = \mathbf{x}[e_{i_2}] = 1$, $\mathbf{x}[e_{i_3}] = 0$, and $x_i = 1$

Any other combination of assignments to the entries of U results in a monochromatic triangle and $T_i \cdot \mathbf{x} \in \{-1, 0, 2, 3\}$. ■

Although initial testing of this method did not find any substantial results, this and similar methods are definitely worth further investigation.

Bibliography

[1] M. Ajtai. Generating Hard Instances of Lattice Problems. In *Proceedings of the 28th Annual ACM Symposium on the Theory of Computing*, 99–108, 1996.

[2] R. B. Allan and R. Laskar. On domination and some related concepts in graph theory. In *9th Southeastern Confernece on Combinatorics, Graph Theory and Computing*, Utilitas Mathematica, 43–56, Winnipeg, 1978.

[3] N. Alon and V. Rödl. Sharp bounds for some multicolor Ramsey numbers. *Combinatorica.* **25** (2005) 125–141.

[4] N. Alon and J. H. Spencer. *The Probabilistic Method.* Wiley-Interscience, New York, 2nd edition, 200.

[5] G. Audemard and L. Simon. Predicting Learnt Clauses Quality in Modern SAT Solver. In *Twenty-first International Joint Conference on Artificial Intelligence (IJCAI'09)*, July 2009. Software available at `https://www.lri.fr/ simon/?page=glucose`.

[6] A. Barg. At the Dawn of the Theory of Codes. *The Mathematical Intelligencer.* **15** (1993) 20–26.

[7] C. Berge. *The Theory of Graphs and its Applications.* Metheun, London, 1962.

[8] N. Biggs. Perfect codes in graphs. *Journal of Combinatorial Theory, Series B.* **15** (1973) 289–296.

[9] B. Bollobás. *Extremal Graph Theory.* Dover Publications, New York, 2004.

[10] R. C. Brewster, E. J. Cockayne, and C. M. Mynhardt. Irredundant Ramsey Numbers for Graphs. *Journal of Graph Theory.* **13** (1989) 283–290.

[11] E. F. Brikell. Solving Low Density Knapsack. In *Advances in Cryptology, Proceedings of Crpto '93*, 25–37. Plenum Press, 1984.

[12] A. E. Brouwer, A. M. Cohen, and A. Neumaier. *Distance-Regular Graphs*, chapter 9.2 Hamming Graphs, 261–267. Springer-Verlag, New York, 1989.

[13] W. G. Brown. On Graphs That Do Not Contain A Thomsen Graph. *Canadian Mathematical Bulletin.* **9** (1966) 281–285.

[14] S. Burer and R. D. Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming (Series B).* **95** (2003) 329–357. Software available at `http://dollar.biz.uiowa.edu/~sburer`.

[15] P. J. Cameron and J. H. van Lint. *Designs, Graphs, Codes and their Links.* London Mathematical Society Student Texts 22. Cambridge University Press, 1997.

[16] Y. Caro, Y. Li, C. C. Rousseau, and Y. Zhang. Asymptotic bounds for some bipartite graph: complete graph Ramsey numbers. *Discrete Mathematics.* **220** (2000) 51–56.

[17] J. W. S. Cassles. *An Introduction to the Geometry of Numbers.* Springer-Verlang, New York, 1971.

[18] G. Chartrand and S. Schuster. On The Existence of Specified Cycles in Complementary Graphs. *Bulletin of the American Mathematical Society.* **77** (1971) 995–998.

[19] F. Chung and R. Graham. *Erdős on Graphs: His Legacy of Unsolved Problems.* A K Peters, Wellesley, Massachusetts, 1998.

[20] F. R. K. Chung, R. Cleve, and P. Dagum. A Note on Constructive Lower Bounds for the Ramsey Numbers $R(3,t)$. *Journal of Combinatorial Theory, Series B.* **57** (1993) 150–155.

[21] V. Chvátal and F. Harary. Generalized Ramsey Theory for Graphs, III. Small Off-Diagonal Numbers. *Pacific Journal of Mathematics.* **41** (1972) 335–345.

[22] M. Clancy. Some Small Ramsey Numbers. *Journal of Graph Theory.* **1** (1977) 89–91.

[23] C. Clapham, A. Flockhart, and J. Sheehan. Graphs Without Four-Cycles. *Journal of Graph Theory.* **13** (1989) 29–47.

[24] A. W. Clifton. Irredundant and Mixed Ramsey Numbers. Master's thesis, Department of Mathematics, East Carolina University, 2012.

[25] E. J. Cockayne. Chessboard Domination Problems. *Annals of Discrete Mathematics.* **48** (1991) 13–20.

[26] E. J. Cockayne and S. T. Hedetniemi. Optimal domination in graphs. *IEEE Transactions on Circuits and Systems.* **22** (1975) 855–857.

[27] E. J. Cockayne, S. T. Hedetniemi, and D. J. Miller. Properties of hereditary hypergraphs and middle graphs. *Canadian Mathematical Bulletin.* **21** (1978) 461–468.

[28] E. J. Cockayne and C. M. Mynhardt. The sequence of upper and lower domination, independence and irredundance numbers of a graph. *Discrete Mathematics.* **122** (1993) 89–102.

[29] G. Cohen, I. Honkala, S. Litsyn, and A. Lobstein. *Covering Codes.* Elsevier, 1997.

[30] C. W. Commander. Maximum Cut Problem, MAX-CUT. In C. Floudas and P. Pardalos, editors, *Encyclopedia of Optimization*, 1991–1999. Springer, second edition, 2009.

[31] S. O. Committee. The international sat competitions web page. `http://www.satcompetition.org/`.

[32] S. Cook. The complexity of theorem proving procedures. In *Third Annual ACM Symposium on Theory of Computing*, 151–158, 1971.

[33] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms.* MIT Press and McGraw-Hill, 3rd edition, 2009.

[34] R. Davies and G. F. Royle. Graph Domination, Tabu Search and the Football Pool Problem. *Discrete Applied Mathematics.* **74** (1995) 217–228.

[35] C. F. de Jaenisch. *Traite des Applications de l'Analyse Mathematiques au jeu des Echecs.* Leningrad, 1862.

[36] A. Dudek and V. Rödl. On the Folkman Number $f(2, 3, 4)$. *Experimental Mathematics.* **17** (2008) 63–67.

[37] P. Erdős. Problems and results in combinatorial analysis *(Italian summary)*. In *Colloquio Internazionale sulle Teorie Combinatorie (Rome, 1973)*, volume 17 of *Tomo II, Atti dei Convegni Lincei*, 3–17, Accademia Nazionale dei Lincei, Rome, 1976.

[38] P. Erdős. On The Combinatorial Problems Which I Would Most Like To See Solved. *Combinatorica.* **1** (1981) 25–42.

[39] P. Erdős, R. J. Faudree, C. C. Rousseau, and R. H. Schelp. On Cycle-Complete Graph Ramsey Numbers. *Journal of Graph Theory.* **2** (1978) 53–64.

[40] P. Erdős and A. Hajnal. Research problem 2–5. *Journal of Combinatorial Theory.* **2** (1967) 104.

[41] P. Erdős and L. Lovász. Problems and results on 3-chromatic hypergraphs and some related questions. In A. Hajnal, R. Rado, and V. T. Sós, editors, *Infinite and Finite Sets (to Paul Erdős on his 60th birthday)*, volume II, 609–627. North-Holland, 1975.

[42] P. Erdős and A. Rényi. On a Problem in the Theory of Graphs *(in Hungarian)*. *Publications of the Mathematical Institute of the Hungarian Academy of Sciences.* **7** (1962) 215–235.

[43] P. Erdős, A. Rényi, and V. T. Sós. On a Problem of Graph Theory. *Studia Scientiarum Mathematicarum Hungarica.* **1** (1966) 215–235.

[44] P. Erdős and M. Simonovits. A Limit Theorem in Graph Theory. *Studia Scientiarum Mathematicarum Hungarica.* **1** (1966) 51–57.

[45] P. Erdős and M. Simonovits. Compactness Results in Extremal Graph Theory. *Combinatorica.* **2** (1982) 275–288.

[46] R. J. Evans, J. R. Pulham, and J. Sheehan. On the number of complete subgraphs contained in certain graphs. *Journal of Combinatorial Theory, Series B.* **30** (1981) 364–371.

[47] G. Exoo. Constructing Ramsey Graphs with a Computer. *Congressus Numerantium.* **59** (1987) 31–36.

[48] R. J. Faudree and R. H. Schelp. Some Problems in Ramsey Theory. In Y. Alavi and D. Lick, editors, *Theory and Applications of Graphs*, volume 642 of *Lecture Notes in Mathematics*, 500–515. Springer Berlin Heidelberg, 1978.

[49] J. Folkman. Graphs with monochromatic complete subgraphs in every edge coloring. *SIAM Journal on Applied Mathematics.* **18** (1970) 19–24.

[50] P. Frankl and V. Rödl. Large triangle-free subgraphs in graphs without K_4. *Graphs and Combinatorics.* **2** (1986) 135–144.

[51] A. Frieze and M. Jerrum. Improved Approximation Algorithms for MAX k-CUT and MAX BISECTION. *Algorithmica.* **18** (1997) 67–81.

[52] Z. Füredi. On the number of edges of quadrilateral-free graphs. *Journal of Combinatorial Theory, Series B.* **68** (1996) 1–6.

[53] Z. Füredi, A. Naor, and J. Verstraëte. On the Turán Number for the Hexagon. *Advances in Mathematics.* **203** (2006) 476–496.

[54] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman, New York, 1979.

[55] B. Gärtner and J. Matousek. *Approximation Algorithms and Semidefinite Programming.* Springer, 2012.

[56] M. Gebser, B. Kaufmann, A. Neumann, and T. Schaub. clasp: A Conflict-Driven Answer Set Solver. In *Logic Programming and Nonmonotonic Reasoning '07*, 260–265. Springer, 2007. Software available at `http://www.cs.uni-potsdam.de/clasp/`.

[57] J. Goedgebeur and S. P. Radziszowski. New Computational Upper Bounds for Ramsey Numbers $R(3, k)$. *The Electronic Journal of Combinatorics.* **20** (2013) 1–28.

[58] M. Goemans and D. Williamson. Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming. *Journal of the ACM.* **42** (1995) 1115–1145.

[59] M. J. E. Golay. Notes on Digital Coding. *Proceedings of the Institute of Radio Engineers.* **37** (1949) 657.

[60] A. W. Goodman. On Sets of Acquaintances and Strangers at Any Party. *The American Mathematical Monthly.* **66** (1959) 778–783.

[61] R. L. Graham. On Egewise 2-Colored Graphs with Monochromatic Triangles and Containing No Complete Hexagon. *Journal of Combinatorial Theory.* **4** (1968) 300.

[62] R. L. Graham and J. H. Spencer. On Small Graphs with Forced Monochromatic Triangles. *Lecture Notes in Mathematics.* **186** (1971) 137–141.

[63] L. Grippo, L. Palagi, M. Piacentini, V. Piccialli, and G. Rinaldi. SpeeDP: An algorithm to compute SDP bounds for very large Max-Cut instances. *Mathematical Programming.* (2012). doi:10.1007/s10107-012-0593-0.

[64] P. M. Gruber and C. G. Lekkerkerker. *Geometry of Numbers.* North-Holland Mathematical Library. North-Holland, 2nd edition, 1987.

[65] B. Grünbaum. *Convex Polytopes*, volume 221 of *Graduate Texts in Mathematics.* Springer-Verlang, New York & London, 2nd edition, 2003.

[66] I. Häcker, M. Lehn, K. Pototzkey, and A. Stippler. FLENS: Flexible Library for Efficient Numerical Solutions. `http://www.mathematik.uni-ulm.de/~lehn/FLENS/`.

[67] N. Hadziivanov and N. Nenov. On the Graham-Spencer number (in Russian). *Comptes rendus de l'Académie bulgare des Sciences.* **32** (1979) 155–158.

[68] H. Hämäläinen, I. Honkala, S. Litsyn, and P. Östergøard. Football Pools–A Game for Mathematicians. **102** (1995) 579–588.

[69] H. Hämäläinen and S. Rankinen. Upper Bounds for Football Pool Problems and Mixed Covering Codes. *Journal of Combinatorial Theory, Series A.* **56** (1991) 84–95.

[70] T. W. Haynes, S. T. Hedetniemi, and P. J. Slater, editors. *Domination in Graphs: Advanced Topics.* Marcel Dekker, Inc., New York, 1998.

[71] T. W. Haynes, S. T. Hedetniemi, and P. J. Slater. *Fundamentals of Domination in Graphs.* Marcel Dekker, Inc., New York, 1998.

[72] C. Helmberg and F. Rendl. A Spectral Bundle Method for Semidefinite Programming. *SIAM Journal of Optimization.* **10** (2000) 673–696. Software available at `http://www-user.tu-chemnitz.de/~helmberg`.

[73] R. Hill and R. W. Irving. On Group Partitions Associated with Lower Bounds for Symmetric Ramsey Numbers. *European Journal of Combinatorics.* **3** (1982) 35–50.

[74] R. Horn and C. Johnson. *Matrix Analysis.* Cambridge University Press, 1985. pp. 176–177.

[75] D. R. Hughes and F. C. Piper. *Projective Planes*, volume 6 of *Graduate Texts in Mathematics.* Springer-Verlang, New York & London, 1973.

[76] R. W. Irving. On a Bound of Graham and Spencer for a Graph-Coloring Constant. *Journal of Combinatorial Theory, Series B.* **15** (1973) 200–203.

[77] M. S. Jacobson and K. Peters. Chordal Graphs and Upper Irredundance, Upper Domination and Independence. *Discrete Mathematics.* **86** (1990) 59–69.

[78] C. J. Jayawardene and C. C. Rousseau. An Upper Bound for the Ramsey Number of a Quadrilateral versus a Complete Graph on Seven Vertices. *Congressus Numerantium.* **130** (1998) 175–188.

[79] C. J. Jayawardene and C. C. Rousseau. The Ramsey Numbers for a Quadrilateral versus All Graphs on Six Vertices. *Journal of Combinatorial Mathematics and Combinatorial Computing.* **35** (2000) 71–87.

[80] L. V. Kantorovich. A new method of solving some classes of extremal problems. *Doklady Akad Sci USSR.* **28** (1940) 211–214.

[81] J. K. Karlof. *Integer Programming: Theory and Practice.* Operations Research Series. CRC Press, 2005.

[82] H. Karloff and U. Zwick. A 7/8 Approximation Algorithm for MAX 3SAT? In *38th Annual IEEE Symposium on Foundations of Computer Science*, 406–415, 1997.

[83] R. M. Karp. Reducibility Among Combinatorial Problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, 85–103. Plenum, New York, 1972.

[84] J. Kelner. *18.409 Topics in Theoretical Computer Science: An Algorithmist's Toolkit, Fall 2009.* (Massachusetts Institute of Technology: MIT OpenCouseWare), Accessed 13 Feb, 2013. `http://ocw.mit.edu`.

[85] G. Kéri. Tables for Bounds on Covering Codes, 2011. (Tables 1, 2, and 5). `http://www.sztaki.hu/~keri/codes/index.htm`.

[86] S. Khot. Hardness of Approximating the Shortest Vector Problem in Lattices. *Journal of the ACM.* **52** (2005) 789–808.

[87] J. H. Kim. The Ramsey Number $R(3,t)$ has Order of Magnitude $t^2/\log t$. *Random Structures and Algorithms.* **7** (1995) 173–207.

[88] W. Kocay. On Writing Isomorphism Programs. In W. Wallis, editor, *Computational and Constructive Design Theory*, 135–175. Kluwer Academic Publishers, 1996.

[89] T. Kövári, V. T. Sós, and P. Turán. On a problem of Zarankiewicz. *Colloquium Mathematicum.* **3** (1954) 50–57.

[90] E. S. Kramer and D. M. Mesner. t-Designs on Hypergraphs. *Discrete Mathematics.* **15** (1976) 263–296.

[91] D. L. Kreher and S. P. Radziszowski. New t-Designs Found by Basis Reduction. In *Proceedings of the 18th Southeastern International Conference on Combinatorics, Graph Theory, and Computing*, volume 59, 155–164. Congressus Numerantium.

[92] D. L. Kreher and S. P. Radziszowski. Finding Simple t-Designs by Using Basis Reduction. *Congressus Numerantium.* **55** (1986) 235–244.

[93] D. L. Kreher and D. R. Stinson. *Combinatorial Algorithms: Generation, Enumeration, and Search.* CRC press LTC, Boca Raton, Florida, 1998.

[94] J. C. Lagarias and A. M. Odlyzko. Solving Low-Density subset sum problems. *Journal of the ACM.* **32** (1985) 229–246.

[95] C. W. H. Lam. The Search for a Finite Projective Plane of order 10. *American Mathematical Monthly.* **98** (1991).

[96] B. A. LaMacchia. Basis Reduction Algorithms and Subset Sum Problems. Master's thesis, Massachusetts Institute of Technology, May 1991.

[97] A. R. Lange. Git repositories for `archer`, `glue`, and `reduce`. `https://github.com/arlange`.

[98] F. Lazebnik, V. A. Ustimenko, and A. J. Woldar. Polarities and $2k$-cycle-free graphs, 1999.

[99] A. K. Lenstra, H. W. Lenstra, and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen.* **261** (1982) 515–534.

[100] H. W. Lenstra. Integer Programming with a Fixed Number of Variables. *Mathematics of Operations Research.* **8** (1983) 538–548.

[101] L. Leonid. Universal search problems *(in Russian). Problems of Information Transmission.* **9** (1973) 384–400.

[102] S. Lin. On Ramsey numbers and K_r-coloring of graphs. *Journal of Combinatorial Theory, Series B.* **12** (1972) 82–92.

[103] J. Linderoth, F. Margot, and G. Thain. Improving Bounds on the Football Pool Problem via Symmetry Reduction and High-Throughput Computing. *INFORMS Journal on Computing.* **21** (2009) 445–457.

[104] I. Livinsky. *Private communication*, 2012.

[105] L. Lu. Explicit Construction of Small Folkman Graphs. *SIAM Journal on Discrete Mathematics.* **21** (2008) 1053–1060.

[106] T. Łuczak, A. Ruciński, and S. Urbański. On minimal Folkman graphs. *Discrete Mathematics.* **236** (2001) 245–262.

[107] F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error-Correcting Codes.* North-Holland Mathematical Library. North-Holland, Amsterdam, 1977.

[108] MATLAB. Version 7.12.0 (R2011a). The MathWorks Inc., Natick, Massachusetts, 2011. `http://www.mathworks.com/products/matlab`.

[109] B. McKay. nauty user's guide (version 2.5). *Australian National University, Department of Computer Science*. Software available at `http://cs.anu.edu.au/~bdm/nauty/`.

[110] B. D. McKay and S. P. Radziszowski. $R(4,5) = 25$. *Journal of Graph Theory.* **19** (1995) 309–322.

[111] D. Micciancio and S. Goldwasser. *Complexity of Lattice Problems: A Cryptographic Perspective*, volume 671 of *The Kluwer International Series in Engineering and Computer Science*. Kluwer Academic Publishers, Boston, Massachusetts, Mar. 2002.

[112] G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization*. Wiley, 1999.

[113] N. Nenov. An example of a 15-vertex $(3,3)$-Ramsey graph with clique number 4 (in Russian). *Comptes rendus de l'Académie bulgare des Sciences.* **34** (1981) 1487–1489.

[114] N. Nenov. On the triangle vertex Folkman numbers. *Discrete Mathematics.* **271** (2003) 327–334.

[115] P. Q. Nguyen and B. V. (Eds.). *The LLL Algorithm: Survey and Applications*. Information Security and Cryptography. Springer, 2009.

[116] Ø. Ore. *Theory of Graphs*, volume 38 of *Colloquium Publications*. American Mathematical Society, Provdence, RI, 1962.

[117] P. R. Östergård and A. Wassermann. A New Lower Bound for the Football Pool Problem for Six Matches. *Journal of Combinatorial Theory, Series A.* **99** (2002) 175–179.

[118] J. Pfaff. *Algorithmic Complexities of Domination-related Graph Parameters*. Ph.D. thesis, Clemson University, 1984.

[119] J. Pfaff, R. Laskar, and S. T. Hedetniemi. NP-completeness of total and connected domination, and irredundnace for bipartite graphs. Technical report, Clemson University, 1984.

[120] K. Piwakowski, S. P. Radziszowski, and S. Urbański. Computation of the Folkman Number $F_e(3,3;5)$. *Journal of Graph Theory.* **32** (1999) 41–49.

[121] S. P. Radziszowski. Small Ramsey Numbers. *Electronic Journal of Combinatorics.* (2011). DS 1, Revision #13. `http://www.combinatorics.org`.

[122] S. P. Radziszowski and D. L. Kreher. Solving Subset Sum Problems with the L^3 Algorithm. *Journal of Combinatorial Mathematics and Combinatorial Computing.* **3** (1988) 49–63.

[123] S. P. Radziszowski and K.-K. Tse. A Computational Approach for the Ramsey Numbers $R(C_4, K_n)$. *Journal of Combinatorial Mathematics and Combinatorial Computing.* **42** (2002) 195–207.

[124] S. P. Radziszowski and X. Xu. On the Most Wanted Folkman Graph. *Geombinatorics.* **16** (2007) 367–381.

[125] F. P. Ramsey. On a Problem of Formal Logic. *Proceedings of London Mathematical Society.* **s2-30** (1930) 264–268.

[126] S. S. Rao. *Engineering Optimization: Theory and Practice.* John Wiley & Sons, Inc, 2009.

[127] F. Rendl, G. Rinaldi, and A. Wiegele. Biq Mac Solver - Binary quadratic and Max cut Solver. `http://biqmac.uni-klu.ac.at/`.

[128] F. Rendl, G. Rinaldi, and A. Wiegele. Solving Max-Cut to Optimality by Intersecting Semidefinite and Polyhedral Relaxations. *Mathematical Programming.* **121** (2009) 307–335.

[129] C. C. Rousseau and C. J. Jayawardene. The Ramsey Number for a Quadrilateral vs. a Complete Graph on Six Vertices. *Congressus Numerantium.* **123** (1997) 97–108.

[130] M. Schaefer. Graph Ramsey Theory and the Polynomial Hierarchy. *Journal of Computer and System Sciences.* **62** (2001) 290–322.

[131] M. Schäuble. Zu einem Kantenfarbungsproblem. Remerkungen zu einer Note von R. L. Graham. *Wiss. Z. Techn. Hochsch. Ilmenau.* **15** (1969) Heft 2 55–58.

[132] Z. Shao, J. Xu, and X. Xu. A New Turán Number for Quadrilateral. *Utilitas Mathematica.* **79** (2009) 51–58.

[133] V. Shoup. Number Theory Library version 5.5.2. `http://www.shoup.net/ntl`.

[134] J. Spencer. Asymptotic Lower Bounds for Ramsey Functions. *Discrete Mathematics.* **20** (1977) 69–76.

[135] J. Spencer. Three hundred million points suffice. *Journal of Combinatorial Theory, Series A.* **49** (1988) 210–217. Also see erratum by M. Hovey in Vol. 50, p. 323.

[136] J. Spencer. Eighty Years of Ramsey $r(3, k)$ ... and Counting! In A. Soifer, editor, *Ramsey Theory: Yesterday, Today and Tomorrow*, volume 285 of *Progress in Mathematics*, 27–39. Springer-Birkhauser, 2011.

[137] W. Stein et al. *Sage Mathematics Software (Version 3.5).* The Sage Development Team, 2012. `http://www.sagemath.org`.

[138] G. Strang. *Introduction to Linear Algebra.* Wellesley-Cambridge Press, Wellesley, Massachusetts, 4th edition, 2009.

[139] H. van Maaren, L. van Norden, and M. J. H. Heule. Sums of squares based approximation algorithms for MAX-SAT. *Discrete Applied Mathematics.* **156** (2008) 1754–1779.

[140] R. J. Vanderbei. *Linear Programming: Foundations and Extensions*, volume 114 of *International Series in Operations Research & Management Science.* Springer-Verlang, 3rd edition, 2008.

[141] X. Xu, Z. Shao, and S. P. Radziszowski. Bounds on Some Ramsey Numbers Involving Quadrilateral. *Ars Combinatoria.* **90** (2009) 337–344.

[142] Y. Yuansheng and P. Rowlinson. On Extremal Graphs Without Four-Cycles. *Utilitas Mathematica.* **41** (1992) 204–210.

[143] G. M. Ziegler. *Lectures on Polytopes*, volume 152 of *Graduate Texts in Mathematics.* Springer-Verlang, Berlin, New York, 1995.